



AST Critical Propulsion and Noise Reduction Technologies for Future Commercial Subsonic Engines

Aeroacoustic Prediction Codes—Supplement: Code Descriptions and Users Guides

*Philip Gliebe, Ramani Mani, Stuart Connell, Samir Salamah, Janet Sober, and Ronald Coffin
General Electric Aircraft Engines, Cincinnati, Ohio*

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., “quick-release” reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to help@sti.nasa.gov
- Fax your question to the NASA STI Information Desk at 757-864-6500
- Telephone the NASA STI Information Desk at 757-864-9658
- Write to:
NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199



AST Critical Propulsion and Noise Reduction Technologies for Future Commercial Subsonic Engines

Aeroacoustic Prediction Codes—Supplement: Code Descriptions and Users Guides

*Philip Gliebe, Ramani Mani, Stuart Connell, Samir Salamah, Janet Sober, and Ronald Coffin
General Electric Aircraft Engines, Cincinnati, Ohio*

Prepared under Contract NAS3–27720

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

Acknowledgments

This report was received by NASA in December 1999 at the completion of the contracted work and describes noise prediction software distributed through the NASA Software Repository.

This work was sponsored by the AST Critical Propulsion and Noise Reduction Technologies for Future Commercial Subsonic Engines.

Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Level of Review: This material has been technically reviewed by NASA expert reviewer(s).

Table of Contents

	<u>Page</u>
1.0 Introduction	1
2.0 Fan Broadband Noise Code Descriptions and Users Guides	3
2.1 Software Requirements	4
2.2 Program Descriptions	4
2.2.1 BBNINPUT – Preprocessor	4
2.2.1.1 Description of Input Files	4
2.2.1.2 Description of Output File	4
2.2.2 ROTOR – Rotor/Turbulence Interaction Noise Program	4
2.2.2.1 Description of Prediction Code	4
2.2.2.2 Description of Main Routine	4
2.2.2.3 Description of Subroutines	5
2.2.2.4 Description of Input File	6
2.2.2.5 Description of Output Files	9
2.2.2.6 ROTOR Source Code Listing	10
2.2.2.7 Sample Run Program ROTOR	89
2.2.2.8 Sample Input File	90
2.2.2.9 Sample Output Files	91
2.2.3 STATOR – Stator / Turbulence Interaction Noise Program	102
2.2.3.1 Description of Prediction Code	102
2.2.3.2 Description of Input File	102
2.2.3.3 Description of Output Files	102
2.2.3.4 STATOR Source Code Listing	102
2.2.3.5 Sample Run Program STATOR	180
2.2.3.6 Sample Input File	180
2.2.3.7 Sample Output Files	182
2.2.4 BBNPLOTS – Postprocessor	192
2.2.4.1 Description of Input Files	193
2.2.4.2 Description of Output Files	193

Table of Contents (Concluded)

	<u>Page</u>
3.0 Fan MPT Superposition Code Descriptions and Users Guides	194
3.1 MPT Prediction System	194
3.2 Software Installation	194
3.3 ROTBLD Users Guide	195
3.4 SUPERPOSE Users Guide	195
4.0 Core Noise Code Description and Users Guide	197
4.1 Software Requirements	197
4.2 Description of Core Noise Code	197
4.3 Description of Core Noise Input File	198
4.4 Description of Core Noise Output File	199
4.5 Core Noise Program Source Code Listing	200
4.6 Sample Input File for Core Noise Program	249
4.7 Sample Output File for Core Noise Program	250
5.0 Empirical Combustor Noise Correlation	262
5.1 COMBUSTOR Source Code	262
5.2 COMBUSTOR Sample Input File	275
5.3 COMBUSTOR Sample Output File	275

1.0 Introduction

This document is a supplement to the final Contract Report prepared by GEAE for the NASA AST (Advanced Subsonic Transport) Technology Noise Reduction Program, NAS3-27720, Area of Interest 13, “Aeroacoustic Prediction Codes.”

The NASA AST Technology Noise Reduction Program has a goal of demonstrating a 10-decibel reduction in EPNL (effective perceived noise level), for aircraft community noise, for several classes of civil aircraft, relative to 1992 technology levels. Of this 10-dB reduction in EPNL, a 6-dB reduction in engine or propulsion system noise is targeted. The remaining reduction is targeted to be demonstrated from reductions in airframe noise, from improvements in aircraft performance, and from defining improved operational (takeoff and landing) procedures. A key ingredient to achieving the 6-dB propulsion system noise reduction goal is having accurate design and analysis tools and codes available that capture the important physics of engine noise generation, propagation, and radiation for each of the significant component noise sources in an aircraft engine. These tools should be capable of carrying out design studies, investigating new concepts for noise reduction, explaining observed results from tests and experiments, and guiding the design of features that will provide the required noise reduction.

The efforts documented herein focused on new and improved models for the various sources of fan “broadband” noise and combustor-related core noise. It was also judged that improvements were needed in the analytic descriptions of the three-dimensional turbulence velocity correlation functions that make up the noise source descriptions for turbulence-generated broadband noise, for both fans and jets. Hence, a task was included to quantify these turbulence correlations for fan broadband noise source model application.

The objective was to establish validated prediction and design analysis tools — methods and codes — applicable to high-bypass commercial turbofans, for: (1) fan broadband noise, (2) fan multiple-pure-tone (MPT) noise, and (3) low-emissions-combustor noise.

The program consisted of four major subtasks:

- Subtask 1 – Improved Aeroacoustic Turbulence Model
- Subtask 2 – Fan Broadband Noise Model
- Subtask 3 – Fan MPT Noise Model
- Subtask 4 – Core Noise Model

Subtask 1 provides improvements in turbulence descriptions and guidance for modeling, feeding into the fan broadband noise model development of subtask 2. Subtasks 1 and 2 combined focus on the eventual objective of reducing broadband noise from high-bypass engines. The third subtask addresses MPT fan noise generated by rotor-bound, shock-wave formations produced when fan rotors operate at supersonic tip speeds. The last subtask addresses possible sources of core noise from new, low-emissions, combustor designs.

The GEAE Program manager was Philip R. Gliebe, and the NASA Contract Technical Manager was Dennis Huff. The fan broadband noise code development was carried out by Dr. Ramani Mani, Dr. Brian Mitchell, Dr. Gregory Ashford, and Ms. Janet Sober of GE Corporate Research and Development (GE-CRD). The Combustor code development was carried out by Dr. Ramani Mani, Dr. Samir Salamah, and Ms. Janet Sober of GE-CRD. The MPT code development work was carried out by

Dr. Stuart Connell of GE–CRD. The directivity model codes were developed by Dr. Edward J. Rice under subcontract to Hersh Acoustical Engineering, Inc. The empirical correlation combustor code was developed by Dr. Jay Mehta and Mr. David Hoskins under subcontract to Diversitec, Inc., and was subsequently streamlined and simplified by Mr. Ronald Coffin of GEAE.

Technical descriptions of the code-development work carried out in subtasks 1 through 4 are documented in a final Contract Report. The resulting computer code descriptions and code users guides are documented in this supplement. These codes include the following:

Section 2.0 – Fan Broadband Noise Code Descriptions and Users Guides

BBNINPUT	Input preprocessor code for running fan broadband noise codes.
ROTOR	Fan rotor-turbulence interaction broadband noise prediction code.
STATOR	Fan stator-turbulence interaction broadband noise prediction code.
BBNPLOTS	Produces plots of output results from ROTOR and STATOR.

Section 3.0 – Fan MPT Superposition Code Descriptions and Users Guides

ROTBLD	Generates a multiple-passage CFD (computational fluid dynamics) mesh system for producing baseline blade geometry perturbation CFD solutions.
SUPERPOSE	Generates complete 360° full-annulus circumferential pressure distribution upstream of a rotor operating at supersonic tip speeds, through superposition of baseline solutions generated by ROTBLD and user-supplied, blade-to-blade distribution of blade geometric variations from the nominal blade shape.

Section 4.0 – Core Noise Code Description and Users Guide

CNOISE	Predicts “indirect” noise produced by combustor temperature fluctuations propagating through downstream turbine stages — using CFD for evaluation of combustor temperature fluctuation parameters and actuator disk theory for propagation through the turbine stages.
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

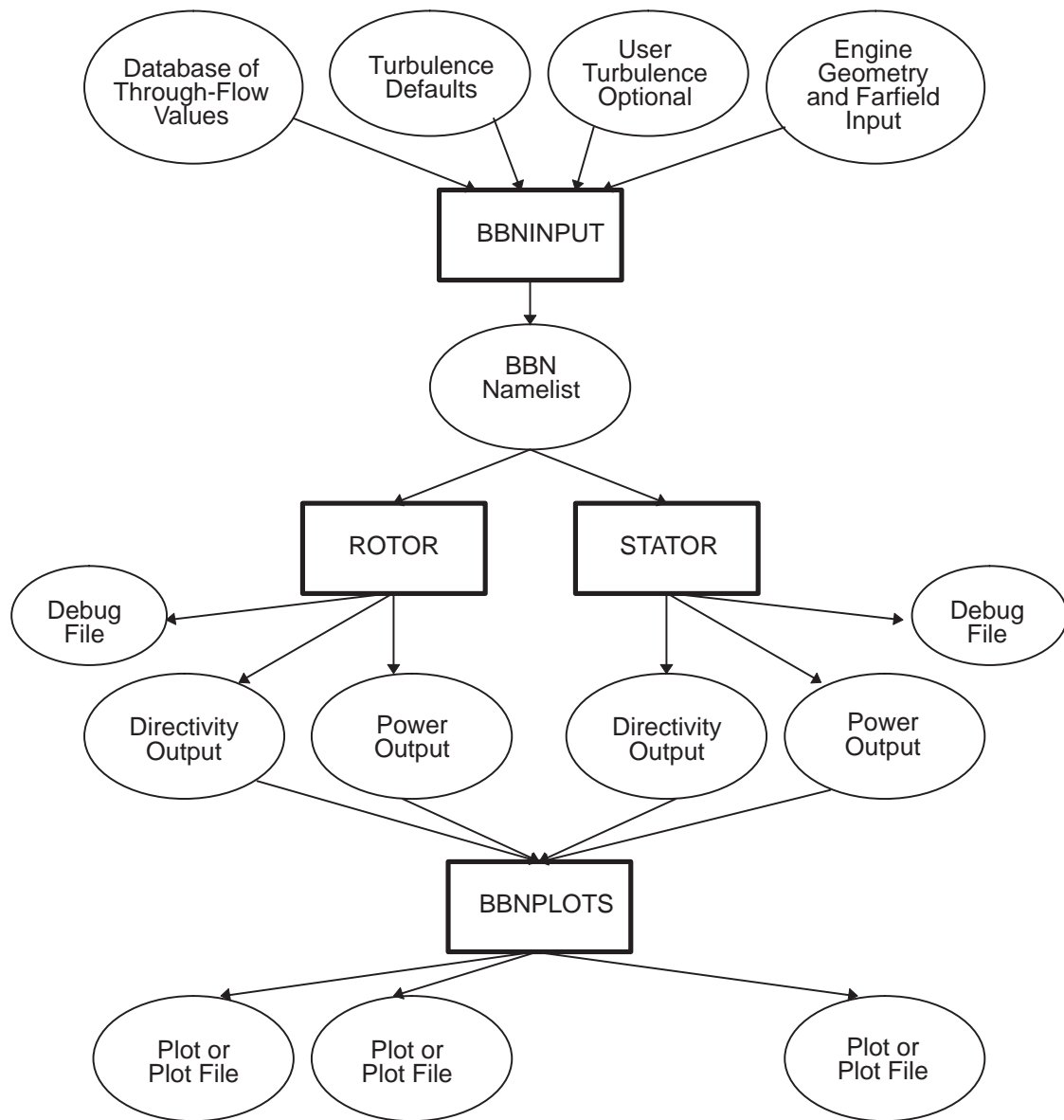
Section 5.0 – Empirical Combustor Noise Correlation

COMBUSTOR	Empirical correlation model prediction of core noise based on GEAE and CFMI engine acoustic data.
-----------	---------------------------------------------------------------------------------------------------

2.0 Fan Broadband Noise Code Descriptions and Users Guides

This section describes software written to support Contract NAS3–27720, Sub AoI 13.2, Fan Broadband Noise Model.

The broadband noise prediction system consists of the broadband noise prediction programs ROTOR and STATOR along with programs BBNINPUT and BBNPLOTS that provide pre- and postprocessing capabilities. The programs are run independently with interaction only through input and output files as shown below.



Broadband Noise Prediction System

2.1 Software Requirements

All programs of the broadband noise prediction system are written in standard Fortran 77.

Programs ROTOR and STATOR must be linked with the IMSL math library from IMSL, Inc. The two routines referenced from that library are QDVAL and SVRGP.

BBNPLOTS (a suggested postprocessor implementation) uses Unix utility XMGR to produce plots. XMGR, an XY plotting tool for workstations or X-terminals using X-windows, may be obtained from website:

<http://plasma-gate.weizmann.ac.il/Xmgr/>

2.2 Program Descriptions

2.2.1 BBNINPUT – Preprocessor

A preprocessor helps to reduce the time and the possibility of manual error when combining through-flow values and turbulence data to create input for programs ROTOR and STATOR. Through-flow values may be obtained from a suitable 3D axisymmetric code and turbulence data may come from test results or from a turbulence model.

BBNINPUT (a suggested preprocessor implementation) gets through-flow values from an in-house (GEAE) code and turbulence values from a previously defined default file. Calculations are performed to produce additional values needed as input for the ROTOR and STATOR programs. The default turbulence data may be overridden by the user. The output file produced is a Fortran namelist format file suitable for input to programs ROTOR and STATOR.

2.2.1.1 Description of Input Files

Input values for through-flow variables are obtained from an in-house code. The default turbulence data input file is a subset of the ROTOR and STATOR input file.

2.2.1.2 Description of Output File

The output file is written as a Fortran namelist file. All variables are listed under the group name INPUT. The complete file description is given with the description of program ROTOR, Section 2.2.2.

2.2.2 ROTOR – Rotor/Turbulence Interaction Noise Program

2.2.2.1 Description of Prediction Code

A listing of the code that yields prediction of broadband noise due to inlet turbulence with the rotor may be found in Section 2.2.2.6, and the line numbers in it will be used to sketch out a description of the code. Salient differences (which are quite minor) of the code used to predict noise due to interaction of inlet turbulence with a rotor and noise due to interaction of inlet turbulence with the outlet guide vane row are indicated.

2.2.2.2 Description of Main Routine

Lines 1 – 156 are preliminary operations. Since the noise from each strip is considered uncorrelated from the other strips, each strip is calculated separately.

In the loop starting at line 162, the process of calculating each strip is started. The acoustic power is normalized initially by the flux of incoming turbulent kinetic energy. Also references will be found to variables with the characters MUG. These arise from the fact that the semiempirical method of Mugridge is also included in the present prediction code.

In lines 270 – 296, the various possibilities of calculating the steady lift coefficient are indicated.

The loop from lines 498 to 533 accomplishes the following. For a given frequency and radial mode order, only a range of tangential wave numbers corresponds to cut-on modes. In this loop, for this range of cut-on modes, several useful modal properties such as the acoustic power of the modes, the degree to which the mode couples to dipoles of a particular orientation, etc., are evaluated.

Starting with line 350, for each frequency band, the 1/3-octave power level contributions (fore and aft) from each strip are calculated.

In the loop beginning at line 418, scattering of an incident shear wave over $2n/d$ where $n = 0, 1, 30$ is considered. The source noncompactness effect as it affects the flowfield associated with steady loading is calculated. Incidentally, lines 210 – 216 calculate the equivalent exponential attenuation factor in case of supersonic inlet relative Mach number. The steps in lines 344 – 354 relate to the idea that if a harmonic of the blade passing frequency (BPF) falls within the band of integration, the power level is calculated by using three points in the band: the two end points and the harmonic of BPF of interest. Otherwise, only the two end points are used. In the loop beginning at 370, the range of tangential wave numbers of incoming shear waves that yield sound waves above cut-off is employed to determine the dipole and quadrupole noise contributions .

2.2.2.3 Description of Subroutines

Subroutine FKCAL – This subroutine calculates the gust response of an isolated airfoil allowing for both compressibility and aspect ratio effects. In the compressible case, separate formulas for low and high frequencies are used.

Subroutine PHICAL – This subroutine calculates the turbulence spectrum function for specified axial and tangential wave numbers. The spectrum function is integrated over the wave number component in the spanwise direction. The subroutine, in principle, allows for specification of a “sudden contraction” following the location at which the turbulence properties are known, although this feature is not used in the present study.

Subroutine EXINT – This subroutine estimates the integral of a monotonically varying function based on its end point values. It is adapted to a situation where the variation of the function could be rapid between the end points.

Subroutine SGN – This subroutine yields “Sgn(x)” where “Sgn = 1” if “x” is greater than or less than zero and “Sgn = 0” if “x = 0.”

Subroutine FRESNL – As indicated, this subroutine computes Fresnel integrals.

Subroutine MUGRIDGE – This subroutine has not been used in the present study. A semiempirical approach has been proposed by Mugridge for broadband noise. The subroutine MUGRIDGE relates to prediction of noise by Mugridge’s suggested formulas.

Subroutine TRANSOGV – This subroutine calculates the transmission loss, in the actuator disk approximation, of a sound wave incident from upstream on an array of unstaggered flat plates. A uniform, purely axial, subsonic flow is assumed for the mean flow.

Subroutine NEWSUB – Sets up tangential and radial mode orders for calculation of the 3D factor.

Subroutine SUB3D – Calculates the 3D factor.

Subroutine SIMP – Calculates term in numerator of the 3D factor.

Subroutine SIMP1 – Calculates normalizing factor for Fourier Bessel analysis.

Subroutine PHIJ – Bessel functions (J) of order “ORD,” argument “ARG.”

Subroutine PHIY – Similar to PHIJ except Y’s.

Subroutine PHIJD – Derivative of Bessel function (J).

Subroutine PHIRYD – Derivative of Y type Bessel function.

Subroutine RJBESL – Program for calculation of Bessel functions (J).

Subroutine RYBESL – Program for calculation of Bessel functions (Y).

Function GAMMA – Function to calculate the gamma function.

Subroutines ABESJ, ABESY, ABESJD, ARBESYD – Routines for calculation of the Bessel functions J, Y and their derivatives for large arguments and arguments less than order.

Subroutine BBRDCFIN – Inlet radiation (directivity) program from E.J. Rice and associated subprograms.

Subroutine BBRDCFEX – Exit radiation (directivity) program from E.J. Rice and associated subprograms.

In the case of noise due to turbulence intercepting the outlet guide vane (OGV) blade row, the computer program is very similar. As far as the main program is concerned, by and large, the minor differences that may be noticed can be easily understood by consideration that the OGV case differs from the rotor case. The difference is that the OGV may be considered to be an unstaggered rotor traveling at zero wheel tip Mach number. Subroutine TRANSROT is used in place of TRANSOGV and calculates the transmission of upstream traveling sound waves impinging on a rotor from the downstream side. The rotor is modeled as a cascade of flat plates that are staggered at an angle $\tan^{-1}(M_t/M_a)$ with respect to the machine axis and carrying uniform flow at a Mach number of $\sqrt{M_a^2 + M_t^2}$ where M_a , M_t are axial and wheel tip Mach numbers. Actuator disk methods are used.

2.2.2.4 Description of Input File

The input file is a Fortran namelist format file with one group named INPUT. The variables within group INPUT are described below:

ANOZRAT	Fan nozzle exit area divided by fan duct area.
ALIP	Major and minor axes of fan inlet lip approximated by an ellipse (inches). For
BLIP	sharp edged lip, make minor axis less than or equal to 0.01 * DTIP.
BW	Bandwidth. This must be set equal to zero to obtain third octave PWL’s.
DELANG	Angular separation in degrees between two successive far field microphones must be such that both 180 and 120. Are divisible by DELANG.

DTIP	Tip diameter of blade row in inches.
ETAFAN	Fan adiabatic efficiency.
GAM	Specific heat ratio.
HTR	Hub to tip ratio of blade row.
IABSOR	If equal to 0, hard walls are assumed. Not equal to 0 is treated wall case.
ISIDELN	Set this equal to 0 for far field microphones on an arc and equal to 1 for farfield microphones on a sideline.
ITL	If equal to 0 means inlet and exhaust termination losses are included. Not equal to 0 means that such losses are neglected.
KASE	Number of cases to be run: I recommend KASE=1 (have not tested other cases).
LEXIT	(Real) exhaust duct length divided by DTIP.
LINLET	Ratio of inlet length (from leading edge of rotor to inlet) to rotor tip diameter.
MACHS	Mach number of surrounding medium (flight Mach number).
NBLADE	Number of rotor blades.
NBSTD	Number of rotor blades for the case for which the turbulence scales are known.
NCOF	Number of cut-off ratio bins to be used in directivity calculations: should not exceed 200.
NF	Number of 1/3-octave bands to be calculated.
NHM	(Ignore) in case of inlet distortion rotor interaction, number of harmonics of BPF to be calculated.
NSTR	Number of streamlines to be computed.
NTOBNI	Ranges from 1 to NF. Information in the NTOBNI'th frequency band is printed for debug purposes.
NVANE	Number of stators (outlet guide vanes).
RADMIC	Radius or sideline distance in feet of far field microphones from center of inlet.
RHO	Inlet density, lbm/ft ³
RPM	Rotor rpm.
SATIR	Rms turbulence of axial component of inlet turbulence incident on rotor normalized by mean axial velocity (for each streamline).
SATIS	Same as above except for turbulence incident on stator (for each streamline), not used in BR or BS programs
SATIW	Similar to above except that here the component parallel to rotor wake (normalized by steady velocity parallel to rotor wake) is referred to (for each streamline). No longer used — superseded by an internal calculation based on rotor drag coefficient.
SAXSP	Axial spacing between rotor trailing edge and stator leading edge normalized by rotor chord (for each streamline).

SCHDR	Rotor chords in inches (for each streamline).
SCHDS	Stator chords in inches (for each streamline).
SCLOPTR	Lift coefficient calculation options for rotor and stator respectively; integer inputs.
SCLOPTS	1 – prescribed CL, 2 – based on work coefficient, 3 – based on pressure ratio, 4 – based on incidence angle. If rotor inlet relative Mach number exceeds unity, regardless of input, SCLOPTR is assumed to be 3.
SCLR	Rotor steady lift coefficient on each streamline. Used only if SCLOPTR is 1 and inlet relative Mach number to rotor is subsonic.
SCLS	Stator steady lift coefficient on each streamline. Used only if SCLOPTS is 1.
SCO	(For each streamline) static speed of sound.
SCONTR	(For each streamline) contraction ratios from where turbulence is measured or
SCONTS	specified to blade row leading edge for three cases of rotor, stator and wake
SCPNTW	Turbulence incident on stator.
SDIA	(For each streamline) streamline diameter in inches.
SELINR	Inlet length scale of incident turbulence normalized by rotor blade pitch for rotor,
SELINS	Stator and wake turbulence incident on stator (for each streamline).
SELINW	SELINW is no longer used as it is superseded by an internal calculation based on rotor drag coefficient. These length scales are integral scales in the axial direction for SELINR, SELINS.
SEMA	Axial Mach number (for each streamline).
SEMT	Tip Mach number (for each streamline).
SINCDR	(For each streamline) incidence angle in degrees (i.e. angle between inflow
SINCDS	direction and stagger line of blade row) for rotor, stator. Note that reference direction is stagger line of blade row.
SPERC	(For each streamline) height or width of stream annulus normalized by annulus outer radius and expressed as a percentage.
SPIMPI	Imaginary part of inlet outer wall specific impedance for various frequencies (used only if IABSOR is not equal to 0).
SPIMPIE	Imaginary part of exhaust outer wall specific impedance for various frequencies (used only if IABSOR is not equal to 0).
SPIMPR	Real part of inlet outer wall specific impedance for various frequencies (used only if IABSOR is not equal to 0).
SPIMPRE	Real part of exhaust outer wall specific impedance for various frequencies (used only if IABSOR is not equal to 0).

SROTC	(For each streamline) drag coefficient of rotor.
SSADIN	Stagger angle of stator in degrees. (For each streamline).
SSCLR	(For each streamline) ratio of turbulence length scales axial/tangential or
SSCLS	Streamwise to cross stream for rotor, stator, rotor wake turbulence on stator
SSCLW	mechanisms.
SSTATCD	(For each streamline) drag coefficient on stator.
STHETA	(For each streamline) ratio of tangential (absolute) velocity downstream of rotor to rotor speed.
STPRIN	(For each streamline) total pressure ratio across rotor.
STVELR	(For each streamline) ratio of tangential turbulent velocity to axial turbulent
STVELS	velocity.
STVELW	
TOBN	One-third-octave band numbers at which NF calculations are desired. Thus if “n” denotes a 1/3-octave band number, the center frequency of the associated band is $10^{(0.1 n)}$.

Variable names IDIST, IPRINT, NDSTLB, SALD, SDELP, SDELU, SPHD, SSIGS, and STHD are still listed in the namelist group in programs ROTOR and STATOR, but these variables are no longer used.

2.2.2.5 Description of Output Files

This section briefly explains some of the outputs the broadband noise codes yield. The following remarks apply to the power and SPL directivity output files.

Practically all outputs in the ROTOR and STATOR power output files are self explanatory. TI stands for turbulence intensity, specifically SATIR for the relevant streamline. CONTR is SCONTR for the relevant streamline. EMR is the tangential length scale (product of RSCAL and the axial length scale). TIT is the tangential turbulence intensity (product of RVEL and TI). AR is the calculated aspect ratio of the blade. The part of the output shown as MUGRIDGE is based on the correlation for total (inlet + exhaust) broadband noise power output suggested by Mugridge.

Concerning STHI, STHUSED, for the subsonic inlet relative Mach numbers, these two outputs are the same and denote the input STHETA. In the supersonic inlet relative Mach number case, STHI is still STHETA but STHUSED is the calculated STHETA consistent with input STPRIN and is the one used to calculate the discharge relative Mach number.

The ROTOR and STATOR SPL directivity output files contain the SPL directivity for each frequency given in the namelist input file. ANGLE is angle from the inlet axis in degrees. SPL is the sound pressure level per header of first output re: $2 \times 10^{-5} \text{ N/m}^2$. The first set is forward radiated, the second set is aft radiated and the third is the total.

2.2.2.6 ROTOR Source Code Listing

```
00001 C      PROGRAM RDIRNEW1.FOR
00002 C*#RUN *=;16130ER/PG/ROTIN2C(BCD,NOGO,CORE=2)
00003 C      ROTIN      ROTOR/TURBULENCE INTERACTION NOISE PREDICTION
00004 C
00005      DIMENSION AEV(46),AED(46),ZMM(46),ZPP(46),DCV1(91),DCD1(91),
00006      &          F(200),DCV2(91),DCD2(91),STHOSR(91),AEVETC(2,91),
00007      &          AEDETC(2,91),TOBN(200),SAXSP(21),SCHDS(21),
00008      &          TROGV(91),F3D(4,200,91),CTFRAT(200,91),
00009      &          CTFRATN(20000),IPERM(20000),CTFRATO(20000),
00010      &          POWTOT(20000),SPIMPR(200),SPIMPI(200),
00011      &          SPIMPRE(200),SPIMPIE(200),FINT(200),SPIMPZ2(200),
00012      &          SPIMPZ2E(200),RICEV(4,200,91),RICED(4,200,91)
00013 C
00014      DIMENSION SAREA(21),SCLR(21),SEMA(21),SEMT(21),SSIGR(21),
00015      &          SSADIN(21),FOB(200),PVT(200),PDT(200),
00016      &          SCONTR(21),SCONTS(21),STVELR(21),SSCLR(21),
00017      &          STPRIN(21),SCO(21),SROTC(21),SDIA(21),SNCD(2,91),
00018      &          SPERC(21),SINCDR(21),SCHDR(21),SCLS(21),SCONTW(21),
00019      &          SINCD(21),SSCLS(21),SSCLW(21),STVELS(21),STVELW(21)
00020 C
00021      DIMENSION DVP(21),DDP(21),QVP(21),QDP(21),CVP(21),CDP(21),
00022      &          SSIGS(21),SDELU(21),NDSTLB(21),SALD(21),STHD(21),
00023      &          SPHD(21),SDELP(21),TVP(21),TDP(21),FJJ(21)
00024 C
00025      DIMENSION SATIR(21),SATIS(21),SATIW(21),SELINR(21),SELINS(21),
00026      &          SELINW(21),SNCU(2,91),SSTATCD(21),STHETA(21)
00027 C
00028      DIMENSION FDVTH(3,200,91),FDDTH(3,200,91),FQVTH(3,200,91),
00029      &          FQDTH(3,200,91),ANGLEO(100)
00030      DIMENSION SDVTH(200,91),SDDTH(200,91),SQVTH(200,91),
00031      &          SQDTH(200,91),STVTH(200,91),STDTH(200,91)
00032      DIMENSION WSUMDV(3,200),WSUMDD(3,200),WSUMQV(3,200),
00033      &          WSUMQD(3,200),WSNDV(3,200),WSNDD(3,200),WSNQV(3,200),
00034      &          WSNQD(3,200),WSDV(200),WSDD(200),WSQV(200),
00035      &          WSTV(200),WSTD(200),ANGLE(100),WSUMIN(200),
00036      &          WSQD(200),WSNDV1(200),WSNDD1(200),WSNQV1(200),
00037      &          WSNQD1(200),SPLVDB(100),SPLDDB(100),SPLOUT(100),
00038      &          SPLVDBT(100,200),SPLDDBT(100,200),SPLDB(100),
00039      &          SPLDBT(100),WUP(200,200),WDN(200,200),SPLTL(100)
00040      REAL      NDVTH(200,91),NDDTH(200,91),NQVTH(200,91),
00041      &          NQDTH(200,91),MACHS
00042      REAL LEXIT,IXP,IXM,NBNP,NBNM,KYS,KTOTP,KTOTM,KR,LEXLOC
00043      REAL MDB(200),ID,LINLET,LSHOCK,LINLOC,LINOVD,LEXOVD
00044      COMPLEX CTU,CTV,CTRM,CTEXP,CGX,CGY,CSYM,CASYM,CDEN,CDEN1,CPART,
00045      &          IM
00046      REAL FREQNCY
00047      INTEGER FHZ,HSTRIG
00048      INTEGER SCLOPTS,SCLOPTR,SCLOPST
00049      LOGICAL CHECK
00050 C
00051      CHARACTER *60 INFILE,OUTFILE,OUTFILE1,PLOTFILE
00052 C
00053 C
00054      NAMELIST/INPUT/
00055      &          BW, DTIP, GAM, HTR, IDIST, IPRINT, KASE, NBLADE,
00056      &          NBSTD, NDSTLB, NF, NHM, NSTR, NVANE, RHO, RPM, SALD,
00057      &          SCHDR,SCLOPTS,SCLOPTR,SCO, SCONTR, SDELP, SDELU,
00058      &          SDIA, SELINR, SEMA, SEMT, SINCDR, SPERC, SPHD, SROTC,
```

```

00059      &          SSCLR, SSIGS, STHD, STHETA, STPRIN, STVELR, TOBN,
00060      &          SATIR, SATIW, SATIS, SCLR, SCLS, SCONTS, SCONTW, SELINS,
00061      &          SELINW, SINCD, SSCLS, SSCLW, STVELS, STVELW, SCHDS, SAXSP,
00062      &          SSADIN, SSTATCD, LEXIT, LINLET, SPIMPR, SPIMPI,
00063      &          SPIMPRE, SPIMPIE, IABSOR, NTOBNI, NCOF, RADMIC, ISIDELN,
00064      &          ALIP, BLIP, MACHS, ANOZRAT, ETAFAN, DELANG, ITL
00065  C      OPEN INPUT FILE
00066  C
00067      WRITE(*,41)
00068      41 FORMAT(' Enter input file name : ', $)
00069      READ(*,42) INFILE
00070      42 FORMAT(A60)
00071  C
00072      OPEN (UNIT=11, STATUS='OLD', FORM='FORMATTED', FILE=INFILE,
00073      & ERR=45)
00074      GO TO 46
00075      45 PRINT *, 'INPUT FILE NOT FOUND'
00076      GO TO 10001
00077  C
00078  C      OPEN OUTPUT FILES
00079  C
00080      46 WRITE(*,47)
00081      47 FORMAT(' Enter output file name : ', $)
00082      READ(*,42) OUTFILE
00083  C
00084      OPEN (UNIT=12, STATUS='NEW', FORM='FORMATTED', FILE=OUTFILE,
00085      & CARRIAGECONTROL='LIST')
00086  C
00087      WRITE(*,48)
00088      48 FORMAT(' Enter 2nd output file name : ', $)
00089      READ(*,42) OUTFILE1
00090  C
00091      OPEN (UNIT=13, STATUS='NEW', FORM='FORMATTED', FILE=OUTFILE1,
00092      & CARRIAGECONTROL='LIST')
00093  C
00094      WRITE(*,49)
00095      49 FORMAT(' Enter spl plot output file name : ', $)
00096      READ(*,42) PLOTFILE
00097  C
00098      OPEN (UNIT=14, STATUS='NEW', FORM='FORMATTED', FILE=PLOTFILE,
00099      & CARRIAGECONTROL='LIST')
00100  C
00101  C      WRITE TO SPL PLOT OUTPUT FILE TO INDICATE ROTOR INFO
00102  C
00103      WRITE(14, '(\'\'ROTOR SPL PLOT OUTPUT FILE\'')')
00104  C
00105  C      READ INPUT DATA
00106  C
00107  C..      CALL FXOPT(12,1,1,0)
00108      KASE=1
00109      NCASE=0
00110      BW=0.0
00111      IM = CMPLX(0.0,1.0)
00112  C      print *, 'input J of interest'
00113  C      READ *, NTOBNI
00114  C
00115  C****      LOOP TO PROCESS ALL CASES
00116  C
00117      DO WHILE (NCASE .LT. KASE)
00118      1      NCASE=NCASE+1

```

```

00119      READ(11,INPUT,ERR=1000,END=9999)
00120      IF(IABSOR.EQ.0) WRITE(12,*)'      HARD WALL ASSUMED'
00121      IF(IABSOR.NE.0) WRITE(12,*)'      TREATED WALL ASSUMED'
00122      FCOF = NCOF
00123      IF(ISIDELN.EQ.0)WRITE(12,*)'MICROPHONE IS ON AN ARC'
00124      IF(ISIDELN.EQ.1)WRITE(12,*)'MICROPHONE IS ON A SIDELINE'
00125      WRITE(12,*)' MICROPHONE DISTANCE IN FEET IS = ',RADMIC
00126      WRITE(12,*)' MACH NUMBER OF SURROUNDING MEDIUM = ',MACHS
00127      WRITE(13,*)'NOZZLE EXIT AREA/DUCT AREA = ',ANOZRAT
00128      WRITE(12,*)'ESTIMATED FAN ADIABATIC EFFICIENCY = ',ETAFAN
00129      IF ( ITL.NE.0 ) WRITE(13,*)' NO DUCT TRANS LOSS ASSUMED'
00130      IF ( ITL.NE.0 ) WRITE(13,*)' '
00131      WRITE(12,*)' '
00132      WRITE(12,108)
00133      WRITE(12,106) NCASE,KASE
00134      NJJ=10
00135  C
00136      CHECK = .TRUE.
00137      PI=3.1415926
00138      BPF=RPM*NBLADE/60.
00139      DO 20 J=1,NF
00140          DO 18 I = 1,NCOF
00141              WUP(I,J) = 0.00
00142              WDN(I,J) = 0.00
00143      18      CONTINUE
00144          DO 19 I = 1,100
00145              SPLVDBT(I,J) = -150.
00146              SPLDDBT(I,J) = -150.
00147              IF ( J.EQ.1 ) SPLDBT(I) = -150.
00148      19      CONTINUE
00149              PDT(J)=-150.
00150              PVT(J)=-150.
00151              MDB(J)=-150.
00152              FOB(J)=10.**(TOBN(J)*0.1)
00153              FINT(J)=FOB(J)/BPF
00154              SPIMPZ2(J) = SPIMPR(J)**2+SPIMPI(J)**2
00155              SPIMPZ2E(J)= SPIMPRE(J)**2+SPIMPIE(J)**2
00156      20      END DO
00157  C
00158  C      INDEX OVER STRIP NUMBER - KJI
00159  C
00160      IF ( NSTR.EQ.1 ) MIDSTR = 1
00161      IF ( NSTR.GT.1 ) MIDSTR = NSTR/2
00162      DO 1949 KJI=1,NSTR
00163          LINLOC = LINLET*DTIP/SDIA(KJI)
00164          LINOVD = FLOAT(NBLADE)*LINLOC/PI
00165          LEXLOC = LEXIT*DTIP/SDIA(KJI)
00166          LEXOVD = FLOAT(NBLADE)*LEXLOC/PI
00167          AR = 0.5*DTIP*(1.-HTR)/SCHDR(KJI)
00168          RAR = AR
00169          PER = SPERC(KJI)
00170          OD = SDIA(KJI)
00171          ID = (1.-0.01*PER)*OD
00172          RUPP = OD/DTIP
00173          IF ( RUPP.GT.1.00 ) RUPP = 1.00
00174          RLOW = RUPP*(1.-0.01*PER)
00175          IF ( RLOW.LT.HTR ) RLOW = HTR
00176          IF ( RLOW.LE.HTR ) RUPP = RLOW/(1.-0.01*PER)
00177          BNEW = DTIP*FLOAT(NBLADE)/(2.*PI*SDIA(KJI))
00178          ANEW = BNEW*HTR

```

```

00179      C          PRINT *, 'BNEW,ANEW ',BNEW,ANEW
00180      A      = PI*(OD**2-ID**2)/576.
00181      CDR=SR0TCD(KJI)
00182      TI=SATIR(KJI)
00183      CONTR=SCONTR(KJI)
00184      SIGR=SCHDR(KJI)*FLOAT(NBLADE)/(PI*SDIA(KJI))
00185      EMA=SEMA(KJI)
00186      EMT=SEMT(KJI)*(1.-0.005*PER)
00187      EMR=SQRT(EMA**2+EMT**2)
00188      SSAD=ATAND(EMT/EMA)-SINCDR(KJI)
00189      C Does this (the following line) make all that much sense ? Who knows!
00190      EL=SELINR(KJI)*FLOAT(NBLADE)/FLOAT(NBSTD)
00191      RSCAL=SSCLR(KJI)
00192      RVEL=STVELR(KJI)
00193      TPR=STPRIN(KJI)
00194      ELT=EL/RSCAL
00195      TIT=TI*RVEL
00196      AAA=A
00197      C=SCO(KJI)
00198      TABS = (C/49.0422)**2
00199      PABS = 53.3*TABS*RHO/144.
00200      IF (KJI.EQ.1) PTOT = PABS*TPR
00201      C
00202      C
00203      C      WRITE INPUT DATA
00204      C
00205      WRITE(12,118) KJI
00206      WRITE(12,120)
00207      WRITE(12,122) EMA,SEMT(KJI),TI,SINCDR(KJI),CONTR,
00208      &          SELINR(KJI)
00209      WRITE(12,124)
00210      WRITE(12,126) GAM,RHO,C,SDIA(KJI),SPERC(KJI),TPR
00211      WRITE(12,125)
00212      WRITE(12,127) RPM,NBLADE,NBSTD,HTR,DTIP,SCHDR(KJI)
00213      WRITE(12,136)
00214      WRITE(12,138) EMR,RSCAL,RVEL,ELT,TIT,AR
00215      WRITE(12,*) ' '
00216      WRITE(12,*) ' '
00217      WRITE(12,*) ' CDROTOR = ',CDR
00218      WRITE(12,*) ' INLET LENGTH/TIP DIAMETER = ',LINLET
00219      C
00220      C
00221      C      PRELIMINARY CONSTANTS AND COEFFICIENTS
00222      C
00223      DBL=130.+4.342945*ALOG(.105*RHO*(C*EMA)**3*TI**2*A)
00224      DBLSPL = 4.342945*ALOG(.105*RHO*(C*EMA)**3*TI**2*A)
00225      WATCON = .105*RHO*(C*EMA)**3*TI**2*A
00226      PI=3.1415926
00227      TPI=2.*PI
00228      G1OV2=(GAM-1.)/2.
00229      G1OVG=(GAM-1.)/GAM
00230      TDGP1=2./(GAM+1.)
00231      GEXP = (GAM+1.)/(2.*(GAM-1.))
00232      T11=TPR**G1OVG-1.
00233      IF (KJI.EQ.1) TTOT = TABS*(1.+T11/ETAFAN)
00234      T12=1.+1./(G1OV2*EMA**2)
00235      CR=EMA/EMR
00236      CR2=CR*CR
00237      SR=EMT/EMR
00238      SR2=SR*SR

```

```

00239      IF (EMR.LT.1.) STHETAR= STHETA(KJI)
00240      IF (EMR.GT.1.) STHETAR= 0.625*(CR2/SR2)*T12*T11
00241      WRITE(12,*) 'STHI,STHUSED ',STHETA(KJI),STHETAR
00242      OMM2 = 1.-EMA**2
00243      SR1MM2=SQRT(1.-EMA**2)
00244      T1 = EMA**2
00245      T2 = (EMT*(1.-STHETAR))**2
00246      EMRE = SQRT(T1+T2)
00247      IF ((EMR.GT.1.).AND.(EMRE.GT.1.))PRINT *,
00248      &      ' INLET & EXIT MREL ARE .GT. 1'
00249      IF ((EMR.GT.1.).AND.(EMRE.GT.1.))PRINT *,
00250      &      'MINLET = ',EMR,' MEXIT = ',EMRE
00251      IF ((EMR.GT.1.).AND.(EMRE.LT.1.))PRINT *,
00252      &      ' INLET & EXIT MREL ARE ',EMR,EMRE
00253      IF ((EMR.GT.1.).AND.(EMRE.LT.1.))WRITE(12,*)
00254      &      ' MRINLET = ',EMR,' MREXIT = ',EMRE
00255      IF ( EMR.LT.1. ) SR1MR2=SQRT(1.-EMR**2)
00256      IF ( EMR.GT.1. ) SR1MR2=SQRT(1.-EMRE**2)
00257      SRCR=SR*CR
00258      TR=EMT/EMA
00259      EMRC=EMR/(5.*CR)
00260      FOMUG=1./(SR*SIGR*20.*CDR)
00261      RMUG=4.*(10.0**(-6))*(RHO/32.17)*(C**3)*SIGR*A*CDR*
00262      &      (EMR**6)
00263      DBMUG=130.+4.342945*ALOG(RMUG)
00264      SCLOPST = SCLOPTR
00265      IF ((EMR.GT.1.).AND.(EMRE.LT.1.))SCLOPST=3
00266      C
00267      C
00268      C      STEADY STATE LIFT COEFFICIENT CALCULATION
00269      C
00270      IF (SCLOPST .EQ. 1) THEN
00271      C
00272      C CL BASED ON INPUT LIFT COEFFICIENTS
00273      C
00274      CL=SCLR(KJI)
00275      ELSE IF (SCLOPST .EQ. 2) THEN
00276      C
00277      C CL BASED ON WORK COEFFICIENT
00278      C
00279      CL=2.0*EMT*STHETAR/(EMR*SIGR)
00280      ELSE IF (SCLOPST .EQ. 3) THEN
00281      C
00282      C CL BASED ON TOTAL PRESSURE RATIO
00283      C
00284      CL=1.25*(CR2*T12*T11/(SIGR*SR))
00285      ELSE
00286      C
00287      C CL BASED ON STAGGER ANGLE
00288      C
00289      SSAR=SSAD*PI/180.
00290      AINC=ABS(ATAN(TR)-SSAR)
00291      CLF1=5.65/(1.0+1.8/AR)
00292      CLF2=1.8+AR
00293      CLF3=1.8+AR*SR1MR2
00294      CLPR=CLF1*CLF2/CLF3
00295      CL=CLPR*AINC
00296      END IF
00297      GP1OV2 = (GAM+1.)/2.
00298      SSAR=SSAD*PI/180.

```

```

00299      AINC=ABS(ATAN(TR)-SSAR)
00300      IF ( EMR.GT.1. ) THEN
00301          EMU  = ASIN(1./EMR)
00302          T1   = EMR**2/(EMR**2-1.)
00303          DEL  = AINC*T1*GP1OV2
00304          LSHOCK = COS(EMU+SSAR)**2/(2.*DEL)
00305          ALU   = LOG(LINOVD/LSHOCK)/LINOVD
00306      ENDIF
00307  C
00308      500      AA=1.0-EMA**2
00309      WRITE(12,140)SCLOPST
00310  C
00311  C      MORE CONSTANTS AND COEFFICIENTS
00312  C
00313          A=CL*SIGR*EMA*EMT/(4.*AA)
00314          B=CL*SIGR*SR1MR2/(4.*AA)
00315          C=CL*SIGR/4.
00316          ALC=TPI*SR1MR2
00317          BETC=TPI
00318          CHIC=BETC*EMT
00319          CHINC = BNEW*(1.-HTR)/(PI*SR1MM2)
00320          DELC=TPI*EMA*EMT
00321          CDP2=(PI*SIGR/2.0)**2
00322          CAE=1./(EMA*SR1MM2)
00323          NINCO2=6
00324          NINCO2P1=NINCO2+1
00325          NINC=2*NINCO2
00326          FNINC=NINC
00327          DELTH=PI/FNINC
00328          THETA=-PI/2.
00329          IMAX=NINC+1
00330          DSNCT=0.5*SIGR/(1.-EMA**2)
00331  C
00332  C
00333  C      CALCULATION OF WAVE PROPAGATION FACTORS
00334  C
00335  C
00336          WRITE(12,110)
00337          IF (BW .LE. 0.0) THEN
00338              WRITE(12,111)
00339              IBW=0
00340          ELSE
00341              WRITE(12,113) BW
00342              IBW=1
00343          END IF
00344          WRITE(12,112)
00345          WRITE(12,114)
00346  C
00347  C
00348  C      INDEX OVER FREQUENCY - J
00349  C
00350          DO 3000 J=1,NF
00351              F(J)=FOB(J)/BPF
00352              ISKIP=0
00353              IF(BW.GT.0.0.OR.F(J).LT.0.5) ISKIP=1
00354  C
00355              IF (ISKIP .NE. 1) THEN
00356                  FHI=1.122462*F(J)
00357                  FLO=0.890899*F(J)
00358                  FJJ(1)=FLO

```

```

00359          FJJ(2)=F(J)
00360          FJJ(3)=FHI
00361          JJMAX=3
00362          ELSE
00363              FJJ(1)=F(J)
00364              JJMAX=1
00365          END IF
00366          CHI=F(J)*CHIC
00367          CHI2=CHI*CHI
00368          NMAXC = CHI*CHINC-0.000001
00369          NMP1C = NMAXC+1
00370          DO IRAD = 1,NMP1C
00371              IRADM1 = IRAD-1
00372              ENR      = FLOAT(IRAD-1)
00373              ENRC     = ENR/CHINC
00374              ENRC2    = ENRC**2
00375              THETA    = -PI/2.
00376              DO I = 1,IMAX
00377                  CTH   = COS(THETA)
00378                  STH   = SIN(THETA)
00379                  CTH2  = CTH**2
00380                  STH2  = STH**2
00381                  IF ((IRAD.GT.1).OR.(I.NE.NINCO2P1)) THEN
00382                      CTF = SQRT (CHI2*STH2+ENRC2*CTH2)
00383                      CTFRAT(IRAD,I) = CHI/CTF
00384                  ELSE
00385                      CTFRAT(IRAD,I) = 1.E+06
00386                  ENDIF
00387                  THETA = THETA+DELTH
00388              ENDDO
00389          ENDDO
00390      C          IF (J.EQ.NTOBNI) WRITE(13,*)' '
00391      C
00392          SDV=0.
00393          SDD=0.
00394          SQV=0.
00395          SQD=0.
00396          DO IJR = 1,NCOF
00397              WSDV(IJR) = 0.00
00398              WSDD(IJR) = 0.00
00399              WSQV(IJR) = 0.00
00400              WSQD(IJR) = 0.00
00401          ENDDO
00402          DO IRAD = 1,NMP1C
00403              DO I = 1,IMAX
00404                  SDVTH(IRAD,I)=0.00
00405                  SDDTH(IRAD,I)=0.00
00406                  SQVTH(IRAD,I)=0.00
00407                  SQDTH(IRAD,I)=0.00
00408              ENDDO
00409          ENDDO
00410          NVAL=30
00411          NVALP1=NVAL+1
00412      C
00413      C
00414      C      INDEX OVER BPF HARMONIC NUMBER    -    N
00415      C
00416          NNCNT      = 0
00417          JJCNT      = 0
00418          DO 2800 NN=1,NVALP1

```

```

00419          N=NN-1
00420          FLTN=N
00421      1580          EN=N
00422      C          IF (ABS(F(J)-EN) .LE. 8.0) THEN
00423          NNCNT = NNCNT+1
00424          ALD1=ALC
00425          ALDN=ABS(EN)*ALC
00426          AL = ALDN
00427          AL2=AL*AL
00428          ALSGNB=AL*SGN(FLTN)*B
00429          BET=EN*BETC
00430          DEL=DELC*EN
00431          RZQ=0.5*TPI*SIGR*ABS(EN)*SR/(1.-EMA**2)
00432          ZQI=0.5*ABS(AL)*CR*SIGR/(1.-EMA**2)
00433          ZQ=SQRT(RZQ**2+ZQI**2)
00434          SNQA=SQRT(PI*ZQ*0.5)
00435          SNQDEN=(1.+SNQA)**2
00436      C
00437      C
00438      C          INITIATE BANDWIDTH SUBDIVISION AND NUMERICAL INTEGRATION
00439      C          FOR 1/3-OCTAVE SPECTRUM CALCULATION
00440      C
00441          IF (ISKIP .NE. 1) THEN
00442          HSTRIG = 0
00443          IF (EN .GT. FLO .AND. EN .LT. FHI) THEN
00444              FJJ(2)=EN
00445              HSTRIG = 1
00446          ELSE
00447              JJCNT = JJCNT+1
00448              END IF
00449      C
00450          END IF
00451      C
00452          FM=FJJ(2)
00453          JJ=1
00454          DO WHILE (JJ .LE. JJMAX)
00455          DO JR = 1,NCOF
00456              WSUMDV(JJ,JR)=0.00
00457              WSUMDD(JJ,JR)=0.00
00458              WSUMQV(JJ,JR)=0.00
00459              WSUMQD(JJ,JR)=0.00
00460          ENDDO
00461          ETARICE = FJJ(JJ)*BPF*DTIP/(12.*SCO(KJI))
00462          CHI=FJJ(JJ)*CHIC
00463          CHI2=CHI*CHI
00464          FVAL=FJJ(JJ)
00465          THE = QDVAL(FVAL,NF,FINT,SPIMPRE,CHECK)
00466          THI = QDVAL(FVAL,NF,FINT,SPIMPR,CHECK)
00467          Z2E = QDVAL(FVAL,NF,FINT,SPIMPZ2E,CHECK)
00468          Z2I = QDVAL(FVAL,NF,FINT,SPIMPZ2,CHECK)
00469          IF ((NNCNT.EQ.1)) THEN
00470              IF ((J.EQ.1).OR.(J.EQ.NF)) THEN
00471                  TRME = SQRT(Z2E-THE**2)
00472                  TRMI = SQRT(Z2I-THI**2)
00473                  TFREQ = FVAL*BPF
00474              ENDIF
00475          ENDIF
00476          SUMDV=0.
00477          SUMDD=0.
00478          SUMQV=0.

```

```

00479          SUMQD=0.
00480          THETA=-PI/2.
00481          NMAX = CHI*CHINC-0.000001
00482          NMP1 = NMAX+1
00483          IF ( JJ.EQ.1 ) NMP1L = NMP1
00484      2345          CONTINUE
00485  C      INDEX OVER RADIAL MODE ORDERS
00486          DO 2700 IRAD = 1,NMP1
00487          ENR = FLOAT(IRAD-1)
00488          ENRC = ENR/CHINC
00489          ENRC2 = ENRC**2
00490          NRAD = IRAD-1
00491          CHIN = SQRT ( CHI2-(ENR/CHINC)**2 )
00492          RCHI = CHI/CHIN
00493          KR = (ENR*PI)/(BNEW*(1.-HTR)*CHIN)
00494          SUMDVR=0.
00495          SUMDDR=0.
00496          SUMQVR=0.
00497          SUMQDR=0.
00498          THETA=-PI/2.
00499          DO 1230 I=1,IMAX
00500          CTH=COS(THETA)
00501          EMAMC=EMA*RCHI-CTH
00502          EMAPC=EMA*RCHI+CTH
00503          IF ( I .LE. (NINCO2+1) ) THEN
00504              ZMM(I)=EMAMC
00505              ZPP(I)=EMAPC
00506              AEV(I)=CAE/(1.+EMA*CTH)**2
00507              AED(I)=CAE/(1.-EMA*CTH)**2
00508          END IF
00509  C
00510          STH=SIN(THETA)
00511          TERM=SR1MM2*STH*CR
00512          TRM =-SR1MM2*STH*SR
00513          DCV1(I)=(EMAPC*SR-TERM)**2
00514          SNCU(1,I)=ABS(EMAPC*CR-TRM)*DSNCT
00515          DCV2(I)=(EMAPC*SR+TERM)**2
00516          SNCU(2,I)=ABS(EMAPC*CR+TRM)*DSNCT
00517          DCD1(I)=(EMAMC*SR-TERM)**2
00518          SNCD(1,I)=ABS(EMAMC*CR-TRM)*DSNCT
00519          DCD2(I)=(EMAMC*SR+TERM)**2
00520          SNCD(2,I)=ABS(EMAMC*CR+TRM)*DSNCT
00521          STHOSR(I)=STH/SR1MM2
00522          TANANG =ABS( STH*SR1MM2/(CTH-EMA*RCHI) )
00523          THETOGV =ATAN(TANANG)
00524          CALL TRANSOGV ( THETOGV,EMA,TR2 )
00525          TROGV(I) = TR2
00526          INDEX=I
00527          IF ( I .GT. (NINCO2+1) ) INDEX=IMAX+1-I
00528          AEVETC(1,I)=AEV( INDEX)*CDP2*DCV1(I)
00529          AEVETC(2,I)=AEV( INDEX)*CDP2*DCV2(I)
00530          AEDETC(1,I)=AED( INDEX)*CDP2*DCD1(I)
00531          AEDETC(2,I)=AED( INDEX)*CDP2*DCD2(I)
00532          THETA=THETA+DELTH
00533      1230          END DO
00534  C
00535  C
00536  C      INDEX OVER (I) FOR INTEGRATION OVER KY
00537  C
00538          THETA=-PI/2.

```

```

00539          DO 2600 I=1,IMAX
00540          CTH  = COS(THETA)
00541          STH  = SIN(THETA)
00542          CTH2 = CTH**2
00543          STH2 = STH**2
00544          IF ((IRAD.GT.1).OR.(I.NE.NINCO2P1)) THEN
00545              CTF  = SQRT (CHI2*STH2+ENRC2*CTH2)
00546              CTFR = CHI/CTF
00547          ELSE
00548              CTFR = 1.E+09
00549          ENDIF
00550          CTFR2 = CTFR**2
00551          IKY = I-(NINCO2+1)
00552          AKX=BET*TR-CHI/EMA
00553          AKY=-BET+CHIN*STHOS(I)
00554          INDEX=I
00555          IF (I .GT. (NINCO2+1)) INDEX=IMAX+1-I
00556          ZM=ZMM(INDEX)*CHIN
00557          ZP=ZPP(INDEX)*CHIN
00558          DO 2310 L1=1,2
00559      C
00560      C      INLET TURBULENCE SPECTRUM CALCULATION
00561      C
00562          CALL PHICAL(AKX,AKY,CONTR,EL,RSCAL,RVEL,
00563      &              PHIXX,PHIXY,PHIYY)
00564          TPHIXY=2.*PHIXY
00565      C
00566      C      DIPOLE CONTRIBUTION
00567      C
00568          OMR=SIGR*(AKX*CR+AKY*SR)/2.
00569          OM=ABS(OMR)
00570          CALL FKCAL(OM,AR,SIGR,ELT,EMR,FK)
00571          SRF=FK*FK
00572          SRF2=2.0*FK
00573          OMR=ABS(OMR)
00574          PHIT=PHIXX*SR2-TPHIXY*SRCR+PHIYY*CR2
00575          PHITS=ABS(PHIT*SRF)
00576      C
00577          IF (L1 .EQ. 2) THEN
00578              SNZ =CHIN*SNCU(2,I)
00579              SNA =SQRT(PI*SNZ*0.5)
00580              SNDEN=(1.+SNA)**2
00581              FNDVM=AEVETC(2,I)*PHITS/SNDEN
00582              SNZ =CHIN*SNCD(2,I)
00583              SNA =SQRT(PI*SNZ*0.5)
00584              SNDEN=(1.+SNA)**2
00585              FNDDM=AEDETC(2,I)*PHITS*TROGV(I)/SNDEN
00586          ELSE
00587              SNZ =CHIN*SNCU(1,I)
00588              SNA =SQRT(PI*SNZ*0.5)
00589              SNDEN=(1.+SNA)**2
00590              FNDVP=AEVETC(1,I)*PHITS/SNDEN
00591              SNZ =CHIN*SNCD(1,I)
00592              SNA =SQRT(PI*SNZ*0.5)
00593              SNDEN=(1.+SNA)**2
00594              FNDDP=AEDETC(1,I)*PHITS*TROGV(I)/SNDEN
00595          END IF
00596      C
00597      C      QUADRUPOLE CONTRIBUTION
00598      C

```

```

00599      2060          ZTERM=ZP
00600          AKXA=AKX*AA
00601          DO 2200 L2=1,2
00602              ZDELAKE=ZTERM-DEL-AKXA
00603              DEN=(AL2+ZDELAKE**2)**2
00604              PART=ZTERM*(ZDELAKE*A-ALSGNB)+C*ZDELAKE*
00605                  &      CHIN*STHOSR(I)*AA
00606              GX=ZTERM*PART
00607              GY=CHIN*STHOSR(I)*PART*AA
00608      TQ=ABS((GX**2*PHIXX+GY**2*PHIYY+GX*GY*TPHIY)/DEN)
00609              IF (L2 .EQ. 2) THEN
00610                  FNQD=TQ*AED(INDEX)*TROGV(I)
00611              ELSE
00612                  FNQV=TQ*AEV(INDEX)
00613              END IF
00614      C
00615          ZTERM=ZM
00616      2200          END DO
00617      C
00618          IF (L1 .EQ. 2) THEN
00619              FNQVM=FNQV
00620              FNQDM=FNQD
00621          ELSE
00622              FNQVP=FNQV
00623              FNQDP=FNQD
00624              AKX=BET*TR+CHI/EMA
00625              ZP=-ZP
00626              ZM=-ZM
00627          END IF
00628      C
00629      2310          END DO
00630      C
00631          FDV=FNDVP+FNDVM
00632          FDD=FNDDP+FNDDM
00633          FQV=FNQVP+FNQVM
00634          FQD=FNQDP+FNQDM
00635      C
00636          IF ((I .EQ. 1) .OR. (I .EQ. IMAX)) THEN
00637              FDV=FDV/2.
00638              FDD=FDD/2.
00639              FQV=FQV/2.
00640              FQD=FQD/2.
00641          END IF
00642      C
00643          AKYN = CHIN*STHOSR(I)*BNEW
00644          IF ((NNCNT.EQ.1).AND.(JJ.NE.2)) THEN
00645              IF(IABSOR.EQ.0)RICEV(JJ,IRAD,I)=1.00
00646              IF(IABSOR.EQ.0)RICED(JJ,IRAD,I)=1.00
00647          IF(IABSOR.NE.0)THEN
00648              CTH = COS(THETA)
00649              STH = SIN(THETA)
00650              ALP = (CTH-RCHI*EMA)/OMM2
00651              ALM = (-CTH-RCHI*EMA)/OMM2
00652              KYS = STH/SR1MM2
00653              KTOTP = (RCHI-EMA*ALP)
00654              KTOTM = (RCHI-EMA*ALM)
00655              CPHXP = ALP/KTOTP
00656              CPHXM = ALM/KTOTM
00657              CPHRP = KR/KTOTP
00658              CPHRM = KR/KTOTM

```

```

00659      DENP  = SQRT(CPHXP**2+CPHRP**2)
00660      DENM  = SQRT(CPHXM**2+CPHRM**2)
00661      CPHYP  = CPHRP/DENP
00662      CPHYP2 = CPHYP**2
00663      SPHYP  = CPHXP/DENP
00664      CPHYM  = CPHRM/DENM
00665      CPHYM2 = CPHYM**2
00666      SPHYM  = CPHXM/DENM
00667      OMSYP  = 1.+EMA*SPHYP
00668      TOMSYP = 2.*OMSYP
00669      OMSYM  = 1.+EMA*SPHYM
00670      TOMSYM = 2.*OMSYM
00671      OMSYP2 = OMSYP**2
00672      OMSYM2 = OMSYM**2
00673      IF ((I.GT.1).AND.(I.LT.IMAX)) THEN
00674          TPSIXP = CPHRP/(CPHXP+EMA)
00675          TPSIXM = CPHRM/(CPHXM+EMA)
00676          NBNP   = LEXOVD*TPSIXP/(1.-HTR)
00677          NBNM   = -LINOVD*TPSIXM/(1.-HTR)
00678          RIXP=(Z2E*CPHYP2*OMSYP2-THE*TOMSYP*CPHYP+1.)/
00679      &          (Z2E*CPHYP2*OMSYP2+THE*TOMSYP*CPHYP+1.)
00680          RIXM=(Z2I*CPHYM2*OMSYM2-THI*TOMSYM*CPHYM+1.)/
00681      &          (Z2I*CPHYM2*OMSYM2+THI*TOMSYM*CPHYM+1.)
00682          IXP = RIXP**NBNP
00683          IXM = RIXM**NBNM
00684      ELSE
00685          IXP = 0.00
00686          IXM = 0.00
00687      ENDIF
00688      RICEV(JJ,IRAD,I) = IXM
00689      RICED(JJ,IRAD,I) = IXP
00690      ENDIF
00691      CALL NEWSUB(RLOW,RUPP,HTR,AKYN,IKY,IRAD,FC)
00692      F3D ( JJ,IRAD,I ) = FC
00693      ENDIF
00694      IF ((JJCNT.EQ.1).AND.(JJ.EQ.2)) THEN
00695          IF(IABSOR.EQ.0)RICEV(JJ,IRAD,I)=1.00
00696          IF(IABSOR.EQ.0)RICED(JJ,IRAD,I)=1.00
00697          IF(IABSOR.NE.0)THEN
00698              CTH = COS(THETA)
00699              STH = SIN(THETA)
00700              ALP = ( CTH-RCHI*EMA)/OMM2
00701              ALM = (-CTH-RCHI*EMA)/OMM2
00702              KYS = STH/SR1MM2
00703              KTOTP = (RCHI-EMA*ALP)
00704              KTOTM = (RCHI-EMA*ALM)
00705              CPHXP = ALP/KTOTP
00706              CPHXM = ALM/KTOTM
00707              CPHRP = KR/KTOTP
00708              CPHRM = KR/KTOTM
00709              DENP  = SQRT(CPHXP**2+CPHRP**2)
00710              DENM  = SQRT(CPHXM**2+CPHRM**2)
00711              CPHYP  = CPHRP/DENP
00712              CPHYP2 = CPHYP**2
00713              SPHYP  = CPHXP/DENP
00714              CPHYM  = CPHRM/DENM
00715              CPHYM2 = CPHYM**2
00716              SPHYM  = CPHXM/DENM
00717              OMSYP  = 1.+EMA*SPHYP
00718              TOMSYP = 2.*OMSYP

```

```

00719      OMSYM = 1.+EMA*SPHYM
00720      TOMSYM = 2.*OMSYM
00721      OMSYP2 = OMSYP**2
00722      OMSYM2 = OMSYM**2
00723      IF ((I.GT.1).AND.(I.LT.IMAX)) THEN
00724          TPSIXP = CPHRP/(CPHXP+EMA)
00725          TPSIXM = CPHRM/(CPHXM+EMA)
00726          NBNP = LEXOVD*TPSIXP/(1.-HTR)
00727          NBNM = -LINOVD*TPSIXM/(1.-HTR)
00728          RIXP=(Z2E*CPHYP2*OMSYP2-THE*TOMSYP*CPHYP+1.)/
00729      &          (Z2E*CPHYP2*OMSYP2+THE*TOMSYP*CPHYP+1.)
00730          RIXM=(Z2I*CPHYM2*OMSYM2-THI*TOMSYM*CPHYM+1.)/
00731      &          (Z2I*CPHYM2*OMSYM2+THI*TOMSYM*CPHYM+1.)
00732          IXP = RIXP**NBNP
00733          IXM = RIXM**NBNM
00734      ELSE
00735          IXP = 0.00
00736          IXM = 0.00
00737      ENDIF
00738      RICEV(JJ,IRAD,I) = IXM
00739      RICED(JJ,IRAD,I) = IXP
00740      ENDIF
00741      CALL NEWSUB(RLOW,RUPP,HTR,AKYN,IKY,IRAD,FC)
00742      F3D ( JJ,IRAD,I ) = FC
00743      ENDIF
00744      IF ((HSTRIG.EQ.1).AND.(JJ.EQ.2)) THEN
00745          IF(IABSOR.EQ.0)RICECV=1.00
00746          IF(IABSOR.EQ.0)RICED=1.00
00747          IF(IABSOR.NE.0)THEN
00748              CTH = COS(THETA)
00749              STH = SIN(THETA)
00750              ALP = ( CTH-RCHI*EMA)/OMM2
00751              ALM = (-CTH-RCHI*EMA)/OMM2
00752              KYS = STH/SR1MM2
00753              KTOTP = (RCHI-EMA*ALP)
00754              KTOTM = (RCHI-EMA*ALM)
00755              CPHXP = ALP/KTOTP
00756              CPHXM = ALM/KTOTM
00757              CPHRP = KR/KTOTP
00758              CPHRM = KR/KTOTM
00759              DENP = SQRT(CPHXP**2+CPHRP**2)
00760              DENM = SQRT(CPHXM**2+CPHRM**2)
00761              CPHYP = CPHRP/DENP
00762              CPHYP2 = CPHYP**2
00763              SPHYP = CPHXP/DENP
00764              CPHYM = CPHRM/DENM
00765              CPHYM2 = CPHYM**2
00766              SPHYM = CPHXM/DENM
00767              OMSYP = 1.+EMA*SPHYP
00768              TOMSYP = 2.*OMSYP
00769              OMSYM = 1.+EMA*SPHYM
00770              TOMSYM = 2.*OMSYM
00771              OMSYP2 = OMSYP**2
00772              OMSYM2 = OMSYM**2
00773              IF ((I.GT.1).AND.(I.LT.IMAX)) THEN
00774                  TPSIXP = CPHRP/(CPHXP+EMA)
00775                  TPSIXM = CPHRM/(CPHXM+EMA)
00776                  NBNP = LEXOVD*TPSIXP/(1.-HTR)
00777                  NBNM = -LINOVD*TPSIXM/(1.-HTR)
00778                  RIXP=(Z2E*CPHYP2*OMSYP2-THE*TOMSYP*CPHYP+1.)/

```

```

00779      &                ( Z2E*CPHY2*OMSY2+THE*TOMSY*CPHY+1. )
00780      RIXM=( Z2I*CPHY2*OMSY2-THI*TOMSY*CPHY+1. ) /
00781      &                ( Z2I*CPHY2*OMSY2+THI*TOMSY*CPHY+1. )
00782      IXP = RIXP**NBNP
00783      IXM = RIXM**NBNM
00784      ELSE
00785      IXP      = 0.00
00786      IXM      = 0.00
00787      ENDIF
00788      RICECV = IXM
00789      RICECD = IXP
00790      ENDIF
00791      CALL NEWSUB (RLOW,RUPP,HTR,AKYN,IKY,IRAD,FC)
00792      F3DF = FC
00793      ENDIF
00794      IF ( (HSTRIG.NE.1).OR.(JJ.NE.2) ) THEN
00795      RICECV = RICEV(JJ,IRAD,I)
00796      RICECD = RICEV(JJ,IRAD,I)
00797      F3DF = F3D ( JJ,IRAD,I )
00798      ENDIF
00799      SUMDVR=SUMDVR+FDV*F3DF*RICECV
00800      SUMDDR=SUMDDR+FDD*F3DF*RICECD
00801      SUMQVR=SUMQVR+( FQV/SNQDEN ) *F3DF*RICECV
00802      SUMQDR=SUMQDR+( FQD/SNQDEN ) *F3DF*RICECD
00803      FDVTH(JJ,IRAD,I) = FDV*F3DF*RICECV
00804      FDDTH(JJ,IRAD,I) = FDD*F3DF*RICECD
00805      FQVTH(JJ,IRAD,I) = FQV*F3DF*RICECV/SNQDEN
00806      FQDTH(JJ,IRAD,I) = FQD*F3DF*RICECD/SNQDEN
00807      TJR1  = (1.-1./CTFR2)
00808      TJR   = FCOF*TJR1
00809      JR    = 1+INT(TJR)
00810      IF ( JR.GT.NCOF ) JR = NCOF
00811      WSUMDV(JJ,JR)=WSUMDV(JJ,JR)+FDVTH(JJ,IRAD,I)
00812      WSUMDD(JJ,JR)=WSUMDD(JJ,JR)+FDDTH(JJ,IRAD,I)
00813      WSUMQV(JJ,JR)=WSUMQV(JJ,JR)+FQVTH(JJ,IRAD,I)
00814      WSUMQD(JJ,JR)=WSUMQD(JJ,JR)+FQDTH(JJ,IRAD,I)
00815      THETA=THETA+DELTH
00816      2600      END DO
00817      SUMDV=SUMDV+SUMDVR
00818      SUMDD=SUMDD+SUMDDR
00819      SUMQV=SUMQV+SUMQVR
00820      SUMQD=SUMQD+SUMQDR
00821      2700      CONTINUE
00822
00823      PROD = CHI2*DELTH
00824      SNDV=PROD*SUMDV
00825      SNDD=PROD*SUMDD
00826      SNQV=SUMQV*DELTH
00827      SNQD=SUMQD*DELTH
00828      DO IJR = 1,NCOF
00829      WSNDV(JJ,IJR)=PROD*WSUMDV(JJ,IJR)
00830      WSNDD(JJ,IJR)=PROD*WSUMDD(JJ,IJR)
00831      WSNQV(JJ,IJR)=DELTH*WSUMQV(JJ,IJR)
00832      WSNQD(JJ,IJR)=DELTH*WSUMQD(JJ,IJR)
00833      ENDDO
00834      DO IRAD = 1,NMP1
00835      DO I = 1,IMAX
00836      FDVTH(JJ,IRAD,I) = PROD*FDVTH(JJ,IRAD,I)
00837      FDDTH(JJ,IRAD,I) = PROD*FDDTH(JJ,IRAD,I)
00838      FQVTH(JJ,IRAD,I) = FQVTH(JJ,IRAD,I)*DELTH

```

```

00839          FQDTH(JJ,IRAD,I) = FQDTH(JJ,IRAD,I)*DELTH
00840          ENDDO
00841      ENDDO
00842      DVP(JJ)=SNDV
00843          DDP(JJ)=SNDD
00844          QVP(JJ)=SNQV
00845          QDP(JJ)=SNQD
00846      DO IRAD = 1,NMP1
00847          DO I = 1,IMAX
00848              NDVTH(IRAD,I)=FDVTH(JJ,IRAD,I)
00849              NDDTH(IRAD,I)=FDDTH(JJ,IRAD,I)
00850              NQVTH(IRAD,I)=FQVTH(JJ,IRAD,I)
00851              NQDTH(IRAD,I)=FQDTH(JJ,IRAD,I)
00852          ENDDO
00853      ENDDO
00854      DO IJR = 1,NCOF
00855          WSNDV1(IJR) = WSNDV(JJ,IJR)
00856          WSNDD1(IJR) = WSNDD(JJ,IJR)
00857          WSNQV1(IJR) = WSNQV(JJ,IJR)
00858          WSNQD1(IJR) = WSNQD(JJ,IJR)
00859      ENDDO
00860  C
00861          IF (ISKIP .NE. 1) THEN
00862              DVP(JJ)=DVP(JJ)/FJJ(JJ)
00863              DDP(JJ)=DDP(JJ)/FJJ(JJ)
00864              QVP(JJ)=QVP(JJ)/FJJ(JJ)
00865              QDP(JJ)=QDP(JJ)/FJJ(JJ)
00866          DO IJR = 1,NCOF
00867              WSNDV(JJ,IJR) = WSNDV(JJ,IJR)/FJJ(JJ)
00868              WSNDD(JJ,IJR) = WSNDD(JJ,IJR)/FJJ(JJ)
00869              WSNQV(JJ,IJR) = WSNQV(JJ,IJR)/FJJ(JJ)
00870              WSNQD(JJ,IJR) = WSNQD(JJ,IJR)/FJJ(JJ)
00871          ENDDO
00872          END IF
00873          IF (ISKIP .NE. 1) THEN
00874              DO IRAD = 1,NMP1
00875                  DO I = 1,IMAX
00876                      FDVTH(JJ,IRAD,I) = FDVTH(JJ,IRAD,I)/FJJ(JJ)
00877                      FDDTH(JJ,IRAD,I) = FDDTH(JJ,IRAD,I)/FJJ(JJ)
00878                      FQVTH(JJ,IRAD,I) = FQVTH(JJ,IRAD,I)/FJJ(JJ)
00879                      FQDTH(JJ,IRAD,I) = FQDTH(JJ,IRAD,I)/FJJ(JJ)
00880                  ENDDO
00881              ENDDO
00882          END IF
00883  C
00884          JJ=JJ+1
00885      END DO
00886  C          IF (JJMAX.EQ.3) NNCNT = 0
00887  C
00888          IF (ISKIP .NE. 1) THEN
00889              IF (JJMAX .EQ. 3) THEN
00890                  CALL EXINT(DVP(1),DVP(2),FLO,FM,SNDV1)
00891                  CALL EXINT(DVP(2),DVP(3),FM,FHI,SNDV2)
00892                  SNDV=SNDV1+SNDV2
00893              DO IJR = 1,NCOF
00894                  W1 = WSNDV(1,IJR)
00895                  W2 = WSNDV(2,IJR)
00896                  W3 = WSNDV(3,IJR)
00897                  CALL EXINT(W1,W2,FLO,FM,R1)
00898                  CALL EXINT(W2,W3,FM,FHI,R2)

```

```

00899          WSNDV1(IJR) = R1+R2
00900      ENDDO
00901      CALL EXINT(DDP(1),DDP(2),FLO,FM,SNDD1)
00902      CALL EXINT(DDP(2),DDP(3),FM,FHI,SNDD2)
00903          SNDD=SNDD1+SNDD2
00904      DO IJR = 1,NCOF
00905          W1 = WSND(1,IJR)
00906          W2 = WSND(2,IJR)
00907          W3 = WSND(3,IJR)
00908          CALL EXINT(W1,W2,FLO,FM,R1)
00909          CALL EXINT(W2,W3,FM,FHI,R2)
00910          WSND1(IJR) = R1+R2
00911      ENDDO
00912      CALL EXINT(QVP(1),QVP(2),FLO,FM,SNQV1)
00913      CALL EXINT(QVP(2),QVP(3),FM,FHI,SNQV2)
00914          SNQV=SNQV1+SNQV2
00915      DO IJR = 1,NCOF
00916          W1 = WSNQV(1,IJR)
00917          W2 = WSNQV(2,IJR)
00918          W3 = WSNQV(3,IJR)
00919          CALL EXINT(W1,W2,FLO,FM,R1)
00920          CALL EXINT(W2,W3,FM,FHI,R2)
00921          WSNQV1(IJR) = R1+R2
00922      ENDDO
00923      CALL EXINT(QDP(1),QDP(2),FLO,FM,SNQD1)
00924      CALL EXINT(QDP(2),QDP(3),FM,FHI,SNQD2)
00925          SNQD=SNQD1+SNQD2
00926      DO IJR = 1,NCOF
00927          W1 = WSNQD(1,IJR)
00928          W2 = WSNQD(2,IJR)
00929          W3 = WSNQD(3,IJR)
00930          CALL EXINT(W1,W2,FLO,FM,R1)
00931          CALL EXINT(W2,W3,FM,FHI,R2)
00932          WSNQD1(IJR) = R1+R2
00933      ENDDO
00934      ELSE
00935          CALL EXINT(DVP(1),DVP(2),FLO,FHI,SNDV)
00936      DO IJR = 1,NCOF
00937          W1 = WSNDV(1,IJR)
00938          W2 = WSNDV(2,IJR)
00939          CALL EXINT(W1,W2,FLO,FHI,R1)
00940          WSNDV1(IJR) = R1
00941      ENDDO
00942          CALL EXINT(DDP(1),DDP(2),FLO,FHI,SNDD)
00943      DO IJR = 1,NCOF
00944          W1 = WSND(1,IJR)
00945          W2 = WSND(2,IJR)
00946          CALL EXINT(W1,W2,FLO,FHI,R1)
00947          WSND1(IJR) = R1
00948      ENDDO
00949          CALL EXINT(QVP(1),QVP(2),FLO,FHI,SNQV)
00950      DO IJR = 1,NCOF
00951          W1 = WSNQV(1,IJR)
00952          W2 = WSNQV(2,IJR)
00953          CALL EXINT(W1,W2,FLO,FHI,R1)
00954          WSNQV1(IJR) = R1
00955      ENDDO
00956          CALL EXINT(QDP(1),QDP(2),FLO,FHI,SNQD)
00957      DO IJR = 1,NCOF
00958          W1 = WSNQD(1,IJR)

```

```

00959             W2 = WSNQD(2,IJR)
00960             CALL EXINT(W1,W2,FLO,FHI,R1)
00961             WSNQD1(IJR) = R1
00962             ENDDO
00963             END IF
00964 C
00965             END IF
00966 C
00967             IF (ISKIP .NE. 1) THEN
00968                 IF (JJMAX .EQ. 3) THEN
00969                     DO IRAD = 1,NMP1
00970                         DO I = 1,IMAX
00971                             DVP(1) = FDVTH(1,IRAD,I)
00972                             DDP(1) = FDDTH(1,IRAD,I)
00973                             QVP(1) = FQVTH(1,IRAD,I)
00974                             QDP(1) = FQDTH(1,IRAD,I)
00975                             DVP(2) = FDVTH(2,IRAD,I)
00976                             DDP(2) = FDDTH(2,IRAD,I)
00977                             QVP(2) = FQVTH(2,IRAD,I)
00978                             QDP(2) = FQDTH(2,IRAD,I)
00979                             DVP(3) = FDVTH(3,IRAD,I)
00980                             DDP(3) = FDDTH(3,IRAD,I)
00981                             QVP(3) = FQVTH(3,IRAD,I)
00982                             QDP(3) = FQDTH(3,IRAD,I)
00983                             CALL EXINT(DVP(1),DVP(2),FLO,FM,SNQV1)
00984                             CALL EXINT(DVP(2),DVP(3),FM,FHI,SNQV2)
00985                             NDVTH(IRAD,I) = SNQV1+SNQV2
00986                             CALL EXINT(DDP(1),DDP(2),FLO,FM,SNDD1)
00987                             CALL EXINT(DDP(2),DDP(3),FM,FHI,SNDD2)
00988                             NDDTH(IRAD,I) = SNDD1+SNDD2
00989                             CALL EXINT(QVP(1),QVP(2),FLO,FM,SNQV1)
00990                             CALL EXINT(QVP(2),QVP(3),FM,FHI,SNQV2)
00991                             NQVTH(IRAD,I) = SNQV1+SNQV2
00992                             CALL EXINT(QDP(1),QDP(2),FLO,FM,SNQD1)
00993                             CALL EXINT(QDP(2),QDP(3),FM,FHI,SNQD2)
00994                             NQDTH(IRAD,I) = SNQD1+SNQD2
00995                         ENDDO
00996                     ENDDO
00997                 ELSE
00998                     DO IRAD = 1,NMP1
00999                         DO I = 1,IMAX
01000                             DVP(1) = FDVTH(1,IRAD,I)
01001                             DDP(1) = FDDTH(1,IRAD,I)
01002                             QVP(1) = FQVTH(1,IRAD,I)
01003                             QDP(1) = FQDTH(1,IRAD,I)
01004                             DVP(2) = FDVTH(2,IRAD,I)
01005                             DDP(2) = FDDTH(2,IRAD,I)
01006                             QVP(2) = FQVTH(2,IRAD,I)
01007                             QDP(2) = FQDTH(2,IRAD,I)
01008                             CALL EXINT(DVP(1),DVP(2),FLO,FHI,DUM)
01009                             NDVTH(IRAD,I) = DUM
01010                             CALL EXINT(DDP(1),DDP(2),FLO,FHI,DUM)
01011                             NDDTH(IRAD,I) = DUM
01012                             CALL EXINT(QVP(1),QVP(2),FLO,FHI,DUM)
01013                             NQVTH(IRAD,I) = DUM
01014                             CALL EXINT(QDP(1),QDP(2),FLO,FHI,DUM)
01015                             NQDTH(IRAD,I) = DUM
01016                         ENDDO
01017                     ENDDO
01018                 END IF

```

```

01019      C
01020              END IF
01021      C
01022              IF (N .GT. 0) THEN
01023                  SDV=SDV+4.*SNDV
01024                  SDD=SDD+4.*SNDD
01025                  SQV=SQV+4.*SNQV
01026                  SQD=SQD+4.*SNQD
01027              DO IJR = 1,NCOF
01028                  WSDV(IJR)=WSDV(IJR)+4.*WSNDV1(IJR)
01029                  WSDD(IJR)=WSDD(IJR)+4.*WSNDD1(IJR)
01030                  WSQV(IJR)=WSQV(IJR)+4.*WSNQV1(IJR)
01031                  WSQD(IJR)=WSQD(IJR)+4.*WSNQD1(IJR)
01032              ENDDO
01033              ELSE
01034                  SDV=SDV+SNDV
01035                  SDD=SDD+SNDD
01036                  SQV=SQV+SNQV
01037                  SQD=SQD+SNQD
01038              DO IJR = 1,NCOF
01039                  WSDV(IJR)=WSDV(IJR)+WSNDV1(IJR)
01040                  WSDD(IJR)=WSDD(IJR)+WSNDD1(IJR)
01041                  WSQV(IJR)=WSQV(IJR)+WSNQV1(IJR)
01042                  WSQD(IJR)=WSQD(IJR)+WSNQD1(IJR)
01043              ENDDO
01044              END IF
01045              IF (N .GT. 0) THEN
01046      C                  DO IRAD = 1,NMP1L
01047      DO IRAD = 1,NMP1
01048          DO I = 1,IMAX
01049              SDVTH(IRAD,I)=SDVTH(IRAD,I)+4.*NDVTH(IRAD,I)
01050              SDDTH(IRAD,I)=SDDTH(IRAD,I)+4.*NDDTH(IRAD,I)
01051              SQVTH(IRAD,I)=SQVTH(IRAD,I)+4.*NQVTH(IRAD,I)
01052              SQDTH(IRAD,I)=SQDTH(IRAD,I)+4.*NQDTH(IRAD,I)
01053          ENDDO
01054      ENDDO
01055          ELSE
01056      C                  DO IRAD = 1,NMP1L
01057      DO IRAD = 1,NMP1
01058          DO I = 1,IMAX
01059              SDVTH(IRAD,I)=SDVTH(IRAD,I)+NDVTH(IRAD,I)
01060              SDDTH(IRAD,I)=SDDTH(IRAD,I)+NDDTH(IRAD,I)
01061              SQVTH(IRAD,I)=SQVTH(IRAD,I)+NQVTH(IRAD,I)
01062              SQDTH(IRAD,I)=SQDTH(IRAD,I)+NQDTH(IRAD,I)
01063          ENDDO
01064      ENDDO
01065          END IF
01066      C
01067      C                  END IF
01068      C
01069      2800      END DO
01070      C
01071          SDV=SDV*EMRC
01072          SDD=SDD*EMRC
01073          SQV=SQV*EMRC
01074          SQD=SQD*EMRC
01075          STV=SDV+SQV
01076          STD=SDD+SQD
01077      DO IJR = 1,NCOF
01078          WSDV(IJR)=WSDV(IJR)*EMRC

```

```

01079          WSDD(IJR)=WSDD(IJR)*EMRC
01080          WSQV(IJR)=WSQV(IJR)*EMRC
01081          WSQD(IJR)=WSQD(IJR)*EMRC
01082          WSTV(IJR)=WSDV(IJR)+WSQV(IJR)
01083          WSTD(IJR)=WSDD(IJR)+WSQD(IJR)
01084      ENDDO
01085      TRMDV1= 0.00
01086      TRMDD1= 0.00
01087      TRMQV1= 0.00
01088      TRMQD1= 0.00
01089      TRMV1 = 0.00
01090      TRMD1 = 0.00
01091      DO I = 1,NCOF
01092          TRMDV1= WSDV(I)+TRMDV1
01093          TRMDD1= WSDD(I)+TRMDD1
01094          TRMQV1= WSQV(I)+TRMQV1
01095          TRMQD1= WSQD(I)+TRMQD1
01096          TRMV1 = WSTV(I)+TRMV1
01097          TRMD1 = WSTD(I)+TRMD1
01098      ENDDO
01099      C      DO IRAD = 1,NMP1L
01100      DO IRAD = 1,NMP1
01101          DO I = 1,IMAX
01102              SDVTH(IRAD,I)=SDVTH(IRAD,I)*EMRC
01103              SDDTH(IRAD,I)=SDDTH(IRAD,I)*EMRC
01104              SQVTH(IRAD,I)=SQVTH(IRAD,I)*EMRC
01105              SQDTH(IRAD,I)=SQDTH(IRAD,I)*EMRC
01106              STVTH(IRAD,I)=SDVTH(IRAD,I)+SQVTH(IRAD,I)
01107              STDTH(IRAD,I)=SDDTH(IRAD,I)+SQDTH(IRAD,I)
01108          ENDDO
01109      ENDDO
01110          FHZ=FOB(J)
01111      FHZZ=FOB(J)
01112          DBNB=0.0
01113      IF (J.EQ.NTOBNI) THEN
01114          WRITE(13,*) ' '
01115          POWMAX = STVTH(1,1)+SDVTH(1,1)
01116          ICOUNT = 0
01117      C      DO IRAD = 1,NMP1L
01118      DO IRAD = 1,NMP1
01119          DO I = 1,IMAX
01120              ICOUNT = ICOUNT+1
01121              CTFRATN ( ICOUNT ) = CTFRAT ( IRAD,I )
01122              POWTOT ( ICOUNT ) = STVTH ( IRAD,I ) +
01123          &              STDTH ( IRAD,I )
01124              IF(POWTOT(ICOUNT).GT.POWMAX)
01125          &              POWMAX = POWTOT( ICOUNT )
01126          ENDDO
01127      ENDDO
01128      DO ISP = 1,ICOUNT
01129          IPERM ( ISP ) = ISP
01130      ENDDO
01131      CALL SVRGP ( ICOUNT,CTFRATN,CTFRATO,IPERM )
01132      DO IRAD = 1,NMP1
01133          DO I = 1,IMAX
01134              CTFINT = CTFRAT ( IRAD,I )
01135              POWINT = STVTH ( IRAD,I ) + STDTH ( IRAD,I )
01136              IF (POWINT.GT.0.00) THEN
01137                  POWDB = 10.00*ALOG10(POWINT/POWMAX)
01138              ELSE

```

```

01139          POWDB = -1.00E+06
01140          ENDIF
01141      C          WRITE(13,*) CTFINT,POWDB
01142          ENDDO
01143      ENDDO
01144      DO ISP = 1,ICOUNT
01145          CTFINT = CTFRATO ( ISP )
01146          IREL = IPERM ( ISP )
01147          POWRAT = POWTOT ( IREL )/ POWMAX
01148          IF (POWRAT.GT.0.00) THEN
01149              POWDB = 10.00*ALOG10(POWRAT)
01150          ELSE
01151              POWDB = -1.00E+06
01152          ENDIF
01153      C          WRITE(13,* ) CTFINT , POWDB
01154          ENDDO
01155          SUMCHKDV = 0.00
01156          SUMCHKDD = 0.00
01157          SUMCHKQV = 0.00
01158          SUMCHKQD = 0.00
01159          SUMCHKV = 0.00
01160          SUMCHKD = 0.00
01161      C          DO IRAD = 1,NMP1L
01162          DO IRAD = 1,NMP1
01163              DO I = 1,IMAX
01164                  SUMCHKDV = SUMCHKDV+SDVTH( IRAD,I)
01165                  SUMCHKDD = SUMCHKDD+SDDTH( IRAD,I)
01166                  SUMCHKQV = SUMCHKQV+SQVTH( IRAD,I)
01167                  SUMCHKQD = SUMCHKQD+SQDTH( IRAD,I)
01168                  SUMCHKV = SUMCHKV+STVTH( IRAD,I)
01169                  SUMCHKD = SUMCHKD+STDTH( IRAD,I)
01170              ENDDO
01171          ENDDO
01172          WRITE(13,*) '          SUMCHK      ',
01173      &          '          SUM'
01174          WRITE(13,*) ' '
01175          WRITE(13,*) 'UPSTR DIPOLE CHECKS ',SUMCHKDV,SDV,TRMDV1
01176          WRITE(13,*) 'DNSTR DIPOLE CHECKS ',SUMCHKDD,SDD,TRMDD1
01177          WRITE(13,*) 'UPSTR QUADRU CHECKS ',SUMCHKQV,SQV,TRMQV1
01178          WRITE(13,*) 'DNSTR QUADRU CHECKS ',SUMCHKQD,SQD,TRMQD1
01179          WRITE(13,*) 'UPSTR TOTAL CHECKS ',SUMCHKV,STV,TRMV1
01180          WRITE(13,*) 'DNSTR TOTAL CHECKS ',SUMCHKD,STD,TRMD1
01181      ENDIF
01182      C          IF (BW .LE. 0.0) THEN
01183          IF (F(J) .LT. 0.5) DBNB= -6.35
01184          ELSE
01185              DBNB=10.0*ALOG10(BW/FOB(J))
01186          END IF
01187          DBNBT = 10.** (0.1*DBNB)
01188      C          SDVDB=10.0*ALOG10(SDV) + DBL + DBNB
01189          SDDDB=10.0*ALOG10(SDD) + DBL + DBNB
01190          SQVDB=10.0*ALOG10(SQV) + DBL + DBNB
01191          SQDDB=10.0*ALOG10(SQD) + DBL + DBNB
01192          STVDB=10.0*ALOG10(STV) + DBL + DBNB
01193          STDDDB=10.0*ALOG10(STD) + DBL + DBNB
01194          IF ( J.EQ.NTOBNI ) WRITE(13,* ) ' OBN = ',TOBN(J)
01195          IF ( J.EQ.NTOBNI ) WRITE(13,* ) ' '
01196          DO IJR = 1,NCOF

```

```

01199          WUP(IJR,J) = WUP(IJR,J)+ DBNBT*WSTV(IJR)*WATCON
01200          WDN(IJR,J) = WDN(IJR,J)+ DBNBT*WSTD(IJR)*WATCON
01201          ENDDO
01202          PVT(J)=10.0*ALOG10(10.0**(PVT(J)/10.0)+
01203          &          10.0**(STVDB/10.0))
01204          PDT(J)=10.0*ALOG10(10.0**(PDT(J)/10.0)+
01205          &          10.0**(STDDDB/10.0))
01206      C
01207          FMUG=F(J)/F0MUG
01208          CALL MUGRIDGE(FMUG,DBMUG,IBW,BW,FHZZ,TBLDB)
01209          MDB(J)=10.0*ALOG10(10.0**(MDB(J)/10.0)+
01210          &          10.0**(TBLDB/10.0))
01211      C
01212          WRITE(12,116) FHZ,F(J),SDVDB,SDDDB,SQVDB,
01213          &          SQDDB,STVDB,STDDDB,TBLDB
01214      C
01215      3000      END DO
01216      C
01217      1949      END DO
01218      C
01219      C
01220      C      FAN TOTAL POWER SPECTRUM
01221      C
01222          WRITE(12,132)
01223          TPR      = STPRIN(MIDSTR)
01224          TABS      = (SCO(MIDSTR)/49.0422)**2
01225          PABS      = 53.3*TABS*RHO/144.
01226          PTOT      = PABS*TPR
01227          T11      = TPR**G10VG-1.
01228          TTOT      = TABS*(1.+T11/ETAFAN)
01229          FMACH      = -ABS(SEMA(MIDSTR))
01230          FMACHS      = -ABS(MACHS)
01231          FMACHD      = ABS(SEMA(MIDSTR))
01232          XM          = FMACHD
01233          XM2          = XM**2
01234          AOAST      = (TDGP1*(1.+G10V2*XM2))*GEXP/XM
01235          ANOZRATC    = 1./AOAST
01236          IF ( ANOZRATC.GT.ANOZRAT ) ANOZRAT = 1.02*ANOZRATC
01237          FMACH2      = ABS(MACHS)
01238          DO 101 J=1,NF
01239      C          FHZ=FOB(J)
01240          ETARICE = FOB(J)*DTIP/(12.*SCO(MIDSTR))
01241          PVDBT=PVT(J)
01242          PDDBT=PDT(J)
01243          PTDB=10.0*ALOG10(10.0**(PVDBT/10.0)+
01244          &          10.0**(PDDBT/10.0))
01245          PTDBM=MDB(J)+3.0103
01246          WRITE(12,134) NINT(TOBN(J)), F(J) ,PVDBT,PDDBT,PTDB,PTDBM
01247          DO IJR = 1,NCOF
01248              WSUMIN(IJR) = WUP (IJR,J)
01249          END DO
01250          CALL BBRDCFIN(TABS,PABS,RADMIC,ISIDELN,DTIP,ALIP,BLIP,FMACH,
01251          &FMACHS,NCOF,WSUMIN,ETARICE,DELANG,NANGLE,ANGLE,
01252          &SPLOUT,SPLTL,WATTS,WATTRAN)
01253          WATTDDB = 130.00 + 10.00*ALOG10(WATTS)
01254          WRITE(13,*)' INPOWER CHECK IN DB',WATTDDB,TOBN(J),PVDBT
01255          NANGI      = NANGLE
01256          DO IANG = 1,NANGI
01257              SPLVDB(IANG) = SPLOUT(IANG)
01258          ENDDO

```

```

01259      DO IJR = 1,NCOF
01260          WSUMIN(IJR) = WDN (IJR,J)
01261      END DO
01262      CALL BBRDCFEX(TTOT,PTOT,TABS,PABS,HTR,ANOZRAT,RADMIC,
01263      & ISIDELN,DTIP,DJET,FMACHD,FMACH1,FMACH2,NCOF,WSUMIN,DELANG,
01264      & ETARICE,NANGLE,ANGLE,SPLOUT,SPLTL,WATTS,WATTRAN,FMACHN,
01265      & COFMIN)
01266      WATTDDB = 130.00 + 10.00*ALOG10(WATTS)
01267      WRITE(13,*)' EXPOWER CHECK IN DB',WATTDDB,TOBN(J),PDDBT
01268      NANG      = NANGLE
01269      DO IANG = 1,NANG
01270          II      = NANGLE+1-IAANG
01271          SPLDDB(IAANG) = SPLOUT(II)
01272          ANGLEO(IAANG) = ANGLE(II)
01273          IF ( IANG.GT.NANGI ) SPLVDB ( IANG ) = -150.00
01274          P2IN      = 10.** (0.1*SPLVDB( IANG ))
01275          P2EX      = 10.** (0.1*SPLDDB( IANG ))
01276          SPLDBT(IAANG) = 10.*ALOG10(P2IN+P2EX)
01277      ENDDO
01278      IF ( J.EQ.NTOBNI ) THEN
01279          DO IANG = 1,NANG
01280              IF ( IANG.EQ.1 ) WRITE(13,*)' TOTAL OVER ALL STRIPS'
01281              IF ( IANG.EQ.1 ) WRITE(13,*)
01282              IF ( IANG.EQ.1 ) WRITE(13,150)
01283              IF ( IANG.EQ.1 ) WRITE(13,*)
01284              WRITE(13,160)ANGLEO( IANG ),SPLVDB( IANG ),
01285      &          SPLDDB( IANG ),SPLDBT( IANG )
01286          ENDDO
01287      ENDIF
01288      C
01289      C      .. Write data to spl plot file
01290      C
01291          FREQNCY = 10.00**(0.1*TOBN(J))
01292          DO IANG = 1,NANG
01293              IF ( IANG.EQ.1 ) WRITE(14,148)INT(FREQNCY),INT(TOBN(J))
01294              IF ( IANG.EQ.1 ) WRITE(14,*)
01295              IF ( IANG.EQ.1 ) WRITE(14,150)
01296              IF ( IANG.EQ.1 ) WRITE(14,*)
01297              WRITE(14,160)ANGLEO( IANG ),SPLVDB( IANG ),
01298      &          SPLDDB( IANG ),SPLDBT( IANG )
01299          ENDDO
01300      C
01301      101      END DO
01302      C
01303      END DO
01304      GO TO 9999
01305      C
01306      C****      ERROR DURING READ
01307      C
01308      1000 WRITE(12,1002)
01309      C
01310      C      FORMAT SECTION
01311      C
01312      106      FORMAT(/,27H                      CASE NUMBER,I4,5H      OF,I4)
01313      108      FORMAT(/16X,23HPROGRAM *** ROTIN2M ***//
01314      &13X,29HRESPONSE OF AN ISOLATED ROTOR
01315      &12X,32HTO INGESTION OF INLET TURBULENCE)
01316      110      FORMAT(/,'      *** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13)',
01317      & ' WATTS ***')
01318      111      FORMAT(/,'                      ONE THIRD OCTAVE')

```

```

01319 112  FORMAT(/,'      FREQUENCY      DIPOLE      QUADRUPOLE',
01320      & '      TOTAL')
01321 113  FORMAT(/,'      BANDWIDTH = ',F6.1,' Hz')
01322 114  FORMAT('      HERTZ      F/BPF      INLET EXHAUST      INLET EXHAUST      INLET',
      & ' EXHAUST MUGRIDGE')
01324 116  FORMAT(I8,F8.4,7F8.1)
01325 118  FORMAT(/24H***** STRIP AREA NUMBER ,I4)
01326 120  FORMAT(/52H      EMA      EMTIP      TI      SINCD      CONTR      L/SSTD)
01327 122  FORMAT(2F9.3,F9.4,F9.2,F8.3,F7.2)
01328 124  FORMAT(/53H      GAM      RHO      C      SDIA      SPERC      TPR)
01329 125  FORMAT(/53H      RPM      NB      NBSTD      HTR      DTIP      CHDR)
01330 126  FORMAT(F9.3,F9.4,F9.0,F9.3,F8.3,F9.3)
01331 127  FORMAT(F9.1,I9,I9,I9,F9.3,F8.3,F9.3)
01332 128  FORMAT(/3X,6H CLW=F08.4,4X,6HCLINC=F08.4,4X,6HCLINP=F08.4)
01333 132  FORMAT(/12X,24HFAN TOTAL POWER SPECTRUM//
01334      &7X,4HTOBN,8X,5HF/BPF,6X,6HPWL-UP,6X,6HPWL-DN,5X,7HPWL-TOT,4X,
01335      &8HMUGR-TOT//)
01336 134  FORMAT(I11,F13.4,4F12.2)
01337 136  FORMAT(/53H      EMR      RSCAL      RVEL      ELT      TIT      AR )
01338 138  FORMAT(F9.3,F9.2,F9.4,F9.3,F9.4,F9.3)
01339 140  FORMAT(/,' STREAMLINE LIFT COEFFICIENT CALCULATED U
01340      &SING SCLOPT= ',I2)
01341 148  FORMAT(/,1X,'FREQUENCY =',I6,', OBN =',I3)
01342 150  FORMAT(1X,'ANGLE',1X,'INL SPL',1X,'EXH SPL',1X,'TOT SPL')
01343 160  FORMAT(1X,F5.1,1X,3(1X,F7.1))
01344 1002 FORMAT(/6X,41H** INPUT ERROR ** PROCEEDING TO NEXT CASE//)
01345 C
01346 9999 CLOSE (UNIT=5)
01347      CLOSE (UNIT=6)
01348 10001 CONTINUE
01349 C
01350      STOP
01351      END

00001 C
00002 C      FKCAL      CALCULATION OF FK
00003 C
00004      SUBROUTINE FKCAL(OM,AR,SIGR,ELT,EMR,FK)
00005 C
00006      PI=3.1415926
00007 C
00008      IF(EMR.GT.0.8) GOTO 15
00009      BETASQ=1.0-EMR*EMR
00010      OMS=OM/BETASQ
00011      AMU=EMR*OMS
00012      IF(AMU.GT.1.0) GOTO 15
00013      BETA=SQRT(BETASQ)
00014      OMK=OMS*EMR*EMR
00015      SEARS=SQRT(1.0/(2.0*PI*OMS+1.0/(1.0+2.4*OMS)))
00016      SEARS=SEARS/BETA
00017      SEARS=SEARS*SQRT(1.0-(0.5*OMK)**2)
00018      GOTO 25
00019 15 CONTINUE
00020      EX=2.0*EMR*OM/(1.0+EMR)
00021      Z =SQRT(2.0*EX/PI)
00022      CALL FRESNL(Z,C2X,S2X)
00023      SEARS=1.0/(PI*OM)
00024      SEARS=SEARS*SQRT(2.0*(C2X**2+S2X**2)/EMR)
00025 25 CONTINUE
00026      ELTOH=ELT/(AR*SIGR)

```

```

00027      A=1.0/(2.0*ELTOH)
00028      ASQ=A*A
00029      BSQ=(AR**2)*(OM**2+2.0/PI**2)
00030      B=SQRT(BSQ)
00031      DEN=BSQ-ASQ
00032      ANUM=0.0
00033      C
00034      IF (A .LE. 25.0) ANUM=EXP(-2.0*A) -1.0
00035      ACON=ANUM/DEN
00036      BNUM=0.0
00037      C
00038      IF (B .LE. 25.0) BNUM=EXP(-2.0*B) -1.0
00039      BCON=BNUM/DEN
00040      FKSQ=1.0/A + 0.5*(BSQ/ASQ)*ACON - 0.5*(A/B)*BCON
00041      AFKSQ=ABS(FKSQ)
00042      FK=SEARS*SQRT(AFKSQ)
00043      C
00044      RETURN
00045      END

00001      C
00002      C          PHICAL          CALCULATION OF PHIXX, PHIXY, AND PHIYY
00003      C
00004      SUBROUTINE PHICAL(AKX,AKY,CONTR,EL,RSCAL,RVEL,PHIXX,PHIXY,PHIYY)
00005      C
00006      PI=3.1415926
00007      RVEL2=RVEL**2
00008      RSCAL2=RSCAL**2
00009      C=CONTR
00010      IF(C.LE.0.0) C=1.0
00011      EPS=1.0/C**3
00012      OMEPS=1.0-EPS
00013      OMEPS2=OMEPS**2
00014      C
00015      ELA=EL
00016      ELT=EL/RSCAL
00017      AK1=AKX*C
00018      AK2=AKY/SQRT(C)
00019      AKNX=AK1*ELA
00020      AKNY=AK2*ELT
00021      AKNX2=AKNX**2
00022      AKNY2=AKNY**2
00023      C
00024      ALT2=1.0+AKNX2+AKNY2
00025      ALT=SQRT(ALT2)
00026      RA5=1.0/ALT**5
00027      A=ALT/ELT
00028      C
00029      FPHIYY=2.0*RVEL2-1.0/RSCAL2
00030      IF (C .LE. 1.01 .AND. C .GE. 0.99) THEN
00031      C
00032      C          ANALYTIC TWO-DIMENSIONAL SPECTRA FOR CONTRACTION RATIO = 1.0
00033      C
00034      CPHIXX=ELT*ELA*RA5/(4.0*PI)
00035      CPHIXY=-3.0*CPHIXX/RSCAL
00036      CPHIYY=CPHIXX
00037      C
00038      PHIXX=CPHIXX*(3.0*AKNY2+ALT2)
00039      PHIXY=CPHIXY*(AKNX*AKNY)
00040      PHIYY=CPHIYY*(3.0*AKNX2/RSCAL2+ALT2*FPHIYY)

```

```

00041      RETURN
00042      END IF
00043      C
00044      C      NUMERICAL INTEGRATION OVER K3 FOR CONTRACTION RATIO .NE. 1.0
00045      C
00046      RC2 =1.0/C**2
00047      RSQC=1.0/SQRT(C)
00048      CPHI=4.0*SQRT(C)*ELA*ELT/PI**2
00049      PHI11=0.0
00050      PHI12=0.0
00051      PHI22=0.0
00052      DELTH=PI/36.
00053      CPHI=CPHI*RA5*DELTH
00054      C
00055      DO 30 I=1,18
00056          FI=I-1
00057          TH=FI*DELTH
00058          CTH4=COS(TH)**4
00059          AK3=A*SIN(TH)/COS(TH)
00060          AKNZ=AK3*ELT
00061          AKNZ2=AKNZ**2
00062          AKNT2=AKNY2+AKNZ2
00063          GAM11=AKNT2
00064          GAM12=-AKNX*AKNY/RSCAL
00065          GAM22=AKNX2/RSCAL2+AKNZ2*FPHIYY
00066          B2=EPS*AK1**2+AK2**2+AK3**2
00067      C
00068          IF (B2 .LT. 1.0E-05) THEN
00069              TERM11=CTH4*GAM11
00070              TERM12=CTH4*GAM12
00071              TERM22=CTH4*GAM22
00072          ELSE
00073              RB2=1.0/B2
00074              B4=B2**2
00075              RB4=1.0/B4
00076              TERM11=CTH4*RC2*GAM11*(1.0+AK1**2*OMEPS*RB2)**2
00077              TERM12=CTH4*RSQC*(GAM12+RB2*OMEPS*AK1*(GAM11*AK2+GAM12*AK1)
00078      &              +RB4*OMEPS2*AK1**3*AK2*GAM11)
00079              TERM22=CTH4*C*(GAM22+2.0*RB2*OMEPS*AK1*AK2*GAM12
00080      &              +RB4*OMEPS2*((AK1*AK2)**2)*GAM11)
00081          END IF
00082      C
00083          IF (I .LE. 1) THEN
00084              TERM11=0.5*TERM11
00085              TERM12=0.5*TERM12
00086              TERM22=0.5*TERM22
00087          END IF
00088      C
00089          PHI11=PHI11+TERM11
00090          PHI12=PHI12+TERM12
00091          PHI22=PHI22+TERM22
00092      30 END DO
00093      C
00094          PHIXX=PHI11*CPHI
00095          PHIXY=PHI12*CPHI
00096          PHIYY=PHI22*CPHI
00097      C
00098      RETURN
00099      END

```

```

00001  C
00002  C      EXINT      SUBROUTINE EXINT  -  EXPONENTIAL CURVE INTEGRATION
00003  C
00004      SUBROUTINE EXINT(Y1,Y2,X1,X2,YINT)
00005  C
00006  C
00007      YMAX=AMAX1(Y1,Y2)
00008      YMIN=AMIN1(Y1,Y2)
00009      DELX=ABS(X1-X2)
00010      YINT=YMAX*DELX
00011      IF (YMIN.EQ.00) YINT = 0.5*YINT
00012      IF (YMIN.EQ.00) GO TO 100
00013  C
00014      DELY=ALOG(YMAX/YMIN)
00015      IF (DELY .GE. 0.01) THEN
00016          COR1=(1.0-YMIN/YMAX)/DELY
00017          X  =SQRT(YMAX/YMIN-1.0)
00018          COR2=(ATAN(X))/X
00019          COR =SQRT(COR1*COR2)
00020          YINT=YINT*COR
00021      END IF
00022  C
00023      100 CONTINUE
00024      RETURN
00025      END

00001  C
00002  C      SGN      SIGN OF A FUNCTION OR PARAMETER
00003  C
00004      FUNCTION SGN(X)
00005  C
00006      IF (X .EQ. 0.) THEN
00007          SGN=0.
00008      ELSE
00009          SGN=X/ABS(X)
00010      END IF
00011  C
00012      RETURN
00013      END

00001  C
00002  C      FRESNL      FRESNEL INTEGRAL FUNCTIONS C(Z) AND S(Z)
00003  C
00004      SUBROUTINE FRESNL(Z,C,S)
00005  C
00006      PI=3.1415926
00007      X =0.5*PI*Z*Z
00008      COSX=COS(X)
00009      SINX=SIN(X)
00010  C
00011      TOP=1.0+0.926*Z
00012      BOT=2.0+Z*(1.792+3.104*Z)
00013      F  =TOP/BOT
00014  C
00015      TOP=1.0
00016      BOT=2.0+Z*(4.142+Z*(3.492+6.670*Z))
00017      G  =TOP/BOT

```

```

00018  C
00019      C = 0.5 + F*SINX - G*COSX
00020      S = 0.5 - F*COSX - G*SINX
00021  C
00022      RETURN
00023      END

00001  C
00002      SUBROUTINE MUGRIDGE ( FMUG,DBMUG,IBW,BW,FHZZ,TBLDB )
00003      RAL2=1./ALOG(2.)
00004      IF(FMUG.LE.1.)TBLDB=DBMUG+3.0*ALOG(FMUG)*RAL2
00005      IF((FMUG.GE.1.).AND.(FMUG.LE.2.))TBLDB=DBMUG
00006      IF((FMUG.GE.2.).AND.(FMUG.LE.4.))TBLDB=DBMUG-3.0*ALOG(.5*FMUG)
00007      &                                *RAL2
00008      IF(FMUG.GE.4.)TBLDB=DBMUG-3.-8.*ALOG(.25*FMUG)*RAL2
00009      IF(IBW.EQ.1)TBLDB=TBLDB+6.3533+4.342945*ALOG(BW/FHZZ)
00010      RETURN
00011      END

00001      SUBROUTINE TRANSOGV ( THETA,M,TR2 )
00002      REAL M
00003      CTH = COS(THETA)
00004      STH = SIN(THETA)
00005      DENTH = 1.+M**2+2.*M*CTH
00006      CPHI = (-CTH*(1.+M**2)-2.*M)/DENTH
00007      SPHI = (1.-M**2)*STH/DENTH
00008      DENPHI = (1.-M**2)*(1.+M*CTH)/DENTH
00009      TR = (CTH-CPHI)*(1.+M*CTH)/(CTH-CPHI+M*(1.-CTH*CPHI))
00010      TR2 = TR**2
00011      RETURN
00012      END

00001  C
00002      SUBROUTINE NEWSUB ( RLOW,RUPP,HTR,AKYN,IKY,NR,F3DB )
00003  C
00004      ord = abs(akyn)*(1.+htr)/2.
00005  C
00006      N = NR-1
00007      CALL SUB3D ( AKYN,IKY,HTR,RLOW,RUPP,N,F3DB )
00008      RETURN
00009      END

00001  C
00002      SUBROUTINE SUB3D ( KY,IKY,SIG,RL,RU,N,f3d1 )
00003      REAL KY,KMN,kmn2,kmns
00004      if ( (iky.eq.0).and.(n.eq.0) ) then
00005          f3d1 = (ru**2-rl**2)/(1.-sig**2)
00006          go to 100
00007      endif
00008      pi = 3.14159265
00009      enc = pi*float(n)
00010      kmn = sqrt ( ky**2+(enc/(1.-sig))**2 )
00011      kmns = kmn*sig
00012      kmn2 = kmn**2
00013      sig2 = sig**2
00014      ord = abs(ky)*(1.+sig)/2.

```

```

00015      call simp ( rl,ru,kmn,sig,ord,cnn )
00016      call simpl ( kmn,sig,ord,dencn1 )
00017      f3d1 = ( 2.*cnn**2/dencn1 )/(ru**2-rl**2)
00018 100 continue
00019      return
00020      end

00001  c
00002      SUBROUTINE SIMP ( RL,RU,KMN,SIG,ORD,RES )
00003      REAL KMN
00004      c # of points for Simpson's rule (minimum value 7:also needs to be odd)
00005      c # of points determined assuming period of "2*pi."This must be
00006  c reflected in rescaling of argument as below
00007      a = kmn*rl
00008      b = kmn*ru
00009      args = kmn*sig
00010      factor = 1./kmn**2
00011      c # of points for Simpson's rule (minimum value 7:also needs to be odd)
00012      pi = 3.14159265
00013      rint = (b-a)
00014      npml = 5.*(rint/pi)
00015      np = 2*(npml/2)+1
00016      if (np.lt.7) np = 7
00017      npml = np-1
00018  c evaluation by Simpson's rule with np points
00019      delx = (b-a)/float(npml)
00020  c multipliers for end,even and odd terms
00021      mulend = 1.00
00022      muleven = 4.00
00023      mulodd = 2.00
00024      sum = 0.00
00025      rargs = args/ord
00026      if ((ord.le.30).or.(rargs.gt.0.9)) then
00027          call phijd ( args,ord,bjd )
00028          call phiryd ( args,ord,rbyd )
00029      endif
00030      if ((ord.gt.30).and.(rargs.le.0.9)) then
00031          call abesjd ( args,ord,bjd )
00032          call arbesyd ( args,ord,rbyd )
00033      endif
00034      do 200 i = 1,np
00035          x = a+float(i-1)*delx
00036          rx = x/ord
00037          if ((ord.le.30).or.(rx.gt.0.9)) then
00038              call phij ( x,ord,bj )
00039              call phiy ( x,ord,by )
00040              trm= by*rbyd
00041          endif
00042          if ((ord.gt.30).and.(rx.le.0.9)) then
00043              call abesj ( x,ord,bj )
00044              call abesyr ( x,args,ord,trm )
00045          endif
00046          y = (bj-trm*bjd)*x
00047          if ((i.eq.1).or.(i.eq.np)) sum = sum+y*mulend
00048          if ((i.ne.1).and.(i.ne.np)) then
00049              idisc = 2*(i/2)-i
00050              if (idisc.ne.0) sum = sum+y*mulodd
00051              if (idisc.eq.0) sum = sum+y*muleven
00052          endif

```

```

00053      200 continue
00054          aints  = delx*sum/3.
00055          res    = aints*factor
00056          return
00057          end

00001      c
00002          SUBROUTINE SIMP1 ( KMN,SIG,ORD,RES )
00003          REAL KMN
00004              c      # of points for Simpson's rule (minimum value 7:also needs to be odd)
00005              c      # of points determined assuming period of "2*pi."This must be
00006              c      reflected in rescaling of argument as below
00007              a      = kmn*sig
00008              b      = kmn
00009              args    = a
00010              factor  = 1./kmn**2
00011              c      # of points for Simpson's rule (minimum value 7:also needs to be odd)
00012              pi      = 3.14159265
00013              rint    = (b-a)
00014              npml    = 5.*(rint/pi)
00015              np      = 2*(npml/2)+1
00016              if (np.lt.7) np = 7
00017              npml    = np-1
00018              c      evaluation by Simpson's rule with np points
00019              delx    = (b-a)/float(npml)
00020              c      multipliers for end,even and odd terms
00021              mulend  = 1.00
00022              muleven = 4.00
00023              mulodd  = 2.00
00024              sum     = 0.00
00025              rargs   = args/ord
00026              if ((ord.le.30).or.(rargs.gt.0.9)) then
00027                  call phijd ( args,ord,bjd )
00028                  call phiryd ( args,ord,rbyd )
00029              endif
00030              if ((ord.gt.30).and.(rargs.le.0.9)) then
00031                  call abesjd ( args,ord,bjd )
00032                  call arbesyd ( args,ord,rbyd )
00033              endif
00034              do 200 i = 1,np
00035                  x      = a+float(i-1)*delx
00036                  rx     = x/ord
00037                  if ((ord.le.30).or.(rx.gt.0.9)) then
00038                      call phij ( x,ord,bj )
00039                      call phiy ( x,ord,by )
00040                      trm = by*rbyd
00041                  endif
00042                  if ((ord.gt.30).and.(rx.le.0.9)) then
00043                      call abesj ( x,ord,bj )
00044                      call abesyr ( x,args,ord,trm )
00045                  endif
00046                  y      = (bj-trm*bjd)**2*x
00047                  if ((i.eq.1).or.(i.eq.np)) sum = sum+y*mulend
00048                  if ((i.ne.1).and.(i.ne.np)) then
00049                      idisc = 2*(i/2)-i
00050                      if (idisc.ne.0) sum = sum+y*mulodd
00051                      if (idisc.eq.0) sum = sum+y*muleven
00052                  endif
00053      200 continue
00054          aints  = delx*sum/3.

```

```

00055      res      = aints*factor
00056      return
00057      end

00001  C
00002      SUBROUTINE PHIJ ( ARG,ORD,BJARG )
00003      DIMENSION BJ(1000)
00004  C      CALCULATES " JORD(ARG) "
00005      NORD      = ORD
00006      NB        = NORD+1
00007      AL        = ORD-FLOAT(NORD)
00008      IF ( NB.GT.1000 ) PRINT *, 'NEED TO INCREASE DIMENSIONS OF',
00009  1          ' BJ IN SUBROUTINE PHIJ BEYOND 1000'
00010      IF ( NB.GT.1000 ) GO TO 200
00011  C      WRITE(12,*) 'RJBESL,ARG,ORD ',ARG,ORD
00012      CALL RJBESL ( ARG,AL,NB,BJ,NCALC )
00013      IF ( NCALC.LT.NB ) PRINT *, ' ERROR IN BJ CALCULATION !-PHIJ'
00014      IF ( NCALC.LT.NB ) PRINT *, 'ORD,ARG = ',ORD,ARG
00015      IF ( NCALC.LT.NB ) GO TO 100
00016      BJARG = BJ(NB)
00017  C      WRITE(12,*) 'RJBESL,RES ',BJARG
00018  100 CONTINUE
00019  200 CONTINUE
00020      return
00021      END

00001  C
00002      SUBROUTINE PHIY ( ARG,ORD,BYARG )
00003      DIMENSION BY(1000)
00004  C      CALCULATES " YORD(ARG) "
00005      NORD      = ORD
00006      NB        = NORD+1
00007      AL        = ORD-FLOAT(NORD)
00008  C      WRITE(12,*) 'RYBESL,ARG,ORD ',ARG,ORD
00009      CALL RYBESL ( ARG,AL,NB,BY,NCALC )
00010      IF ( NCALC.LT.NB ) PRINT *, ' ERROR IN BY CALCULATION !-PHIY'
00011      IF ( NCALC.LT.NB ) PRINT *, 'ORD,ARG = ',ORD,ARG
00012      IF ( NCALC.LT.NB ) GO TO 100
00013      BYARG = BY(NB)
00014  C      WRITE(12,*) 'RYBESL,RES ',BYARG
00015  100 CONTINUE
00016  200 CONTINUE
00017      return
00018      END

00001  C
00002      SUBROUTINE PHIJD ( ARG,ORD,BJDER )
00003      DIMENSION BJ(1000)
00004  C      CALCULATES " JORD'(ARG) "
00005      NORD      = ORD
00006      NB        = NORD+1
00007      NBP1      = NB+1
00008      IF ( NBP1.GT.1000 ) PRINT *, 'NEED TO INCREASE DIMENSIONS OF',
00009  1          ' BJ IN SUBROUTINE PHIJD BEYOND 1000'
00010      IF ( NBP1.GT.1000 ) GO TO 200
00011      AL        = ORD-FLOAT(NORD)
00012  C      WRITE(12,*) 'RJBESL,ARG,ORD ',ARG,(ORD+1.)
00013      CALL RJBESL ( ARG,AL,NBP1,BJ,NCALC )

```

```

00014      IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN BJDER CALCULATION ! '
00015      IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN PHIJD ! '
00016      IF ( NCALC.LT.NBP1 ) PRINT *, ' ORD,ARG= ', ORD, ARG
00017      IF ( NCALC.LT.NBP1 ) GO TO 100
00018      BJDER = -BJ(NBP1)+ORD*BJ(NB)/ARG
00019 C      WRITE(12,*) 'RJBESL,RES ', BJ(NBP1)
00020      100 CONTINUE
00021      200 CONTINUE
00022      return
00023      END

```

```

00001 C
00002      SUBROUTINE PHIRYD ( ARG,ORD,RBYDER )
00003      DIMENSION BY(1000)
00004 C      CALCULATES " JORD'(ARG) AND YORD'(ARG) "
00005      NORD = ORD
00006      NB = NORD+1
00007      NBP1 = NB+1
00008      IF ( NBP1.GT.1000 ) PRINT *, 'NEED TO INCREASE DIMENSIONS OF',
00009      1      ' BY IN SUBROUTINE PHIYD BEYOND 1000'
00010      IF ( NBP1.GT.1000 ) GO TO 200
00011      AL = ORD-FLOAT(NORD)
00012 C      WRITE(12,*) 'RYBESL,ARG,ORD ', ARG, (ORD+1.)
00013      CALL RYBESL ( ARG,AL,NBP1,BY,NCALC )
00014      IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN BYDER CALCULATION ! '
00015      IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN PHIYD ! '
00016      IF ( NCALC.LT.NBP1 ) PRINT *, ' ORD,ARG = ', ORD, ARG
00017      IF ( NCALC.LT.NBP1 ) GO TO 100
00018      BYDER = -BY(NBP1)+ORD*BY(NB)/ARG
00019      RBYDER = 1./BYDER
00020 C      WRITE(12,*) 'RYBESL,RES ', BY(NBP1)
00021      100 CONTINUE
00022      200 CONTINUE
00023      return
00024      END

```

```

00001 C
00002      SUBROUTINE RJBESL(X, ALPHA, NB, B, NCALC)
00003 C-----
00004 C This routine calculates Bessel functions J sub(N+ALPHA) (X)
00005 C for non-negative argument X, and non-negative order N+ALPHA.
00006 C
00007 C
00008 C Explanation of variables in the calling sequence.
00009 C
00010 C X      - working precision non-negative real argument for which
00011 C          J's are to be calculated.
00012 C ALPHA - working precision fractional part of order for which
00013 C          J's or exponentially scaled J'r (J*exp(X)) are
00014 C          to be calculated.  0 <= ALPHA < 1.0.
00015 C NB    - integer number of functions to be calculated, NB > 0.
00016 C          The first function calculated is of order ALPHA, and the
00017 C          last is of order (NB - 1 + ALPHA).
00018 C B      - working precision output vector of length NB.  If RJBESL
00019 C          terminates normally (NCALC=NB), the vector B contains the
00020 C          functions J/ALPHA/(X) through J/NB-1+ALPHA/(X), or the
00021 C          corresponding exponentially scaled functions.
00022 C NCALC - integer output variable indicating possible errors.
00023 C          Before using the vector B, the user should check that

```

```

00024 C          NCALC=NB, i.e., all orders have been calculated to
00025 C          the desired accuracy.  See Error Returns below.
00026 C
00027 C
00028 C*****
00029 C*****
00030 C
00031 C  Explanation of machine-dependent constants
00032 C
00033 C    it      = Number of bits in the mantissa of a working precision
00034 C             variable
00035 C    NSIG     = Decimal significance desired.  Should be set to
00036 C             INT(LOG10(2)*it+1).  Setting NSIG lower will result
00037 C             in decreased accuracy while setting NSIG higher will
00038 C             increase CPU time without increasing accuracy.  The
00039 C             truncation error is limited to a relative error of
00040 C             T=.5*10**(-NSIG).
00041 C    ENTEN    = 10.0 ** K, where K is the largest integer such that
00042 C             ENTEN is machine-representable in working precision
00043 C    ENSIG    = 10.0 ** NSIG
00044 C    RTNSIG   = 10.0 ** (-K) for the smallest integer K such that
00045 C             K .GE. NSIG/4
00046 C    ENMTEN   = Smallest ABS(X) such that X/4 does not underflow
00047 C    XLARGE   = Upper limit on the magnitude of X.  If ABS(X)=N,
00048 C             then at least N iterations of the backward recursion
00049 C             will be executed.  The value of 10.0 ** 4 is used on
00050 C             every machine.
00051 C
00052 C
00053 C    Approximate values for some important machines are:
00054 C
00055 C
00056 C             it      NSIG      ENTEN      ENSIG
00057 C
00058 C  CRAY-1      (S.P.)    48      15      1.0E+2465  1.0E+15
00059 C  Cyber 180/855
00060 C    under NOS  (S.P.)    48      15      1.0E+322  1.0E+15
00061 C  IEEE (IBM/XT,
00062 C    SUN, etc.) (S.P.)    24       8      1.0E+38   1.0E+8
00063 C  IEEE (IBM/XT,
00064 C    SUN, etc.) (D.P.)    53      16      1.0D+308  1.0D+16
00065 C  IBM 3033    (D.P.)    14       5      1.0D+75   1.0D+5
00066 C  VAX         (S.P.)    24       8      1.0E+38   1.0E+8
00067 C  VAX D-Format (D.P.)    56      17      1.0D+38   1.0D+17
00068 C  VAX G-Format (D.P.)    53      16      1.0D+307  1.0D+16
00069 C
00070 C
00071 C             RTNSIG      ENMTEN      XLARGE
00072 C
00073 C  CRAY-1      (S.P.)    1.0E-4    1.84E-2466  1.0E+4
00074 C  Cyber 180/855
00075 C    under NOS  (S.P.)    1.0E-4    1.25E-293  1.0E+4
00076 C  IEEE (IBM/XT,
00077 C    SUN, etc.) (S.P.)    1.0E-2    4.70E-38   1.0E+4
00078 C  IEEE (IBM/XT,
00079 C    SUN, etc.) (D.P.)    1.0E-4    8.90D-308  1.0D+4
00080 C  IBM 3033    (D.P.)    1.0E-2    2.16D-78   1.0D+4
00081 C  VAX         (S.P.)    1.0E-2    1.17E-38   1.0E+4
00082 C  VAX D-Format (D.P.)    1.0E-5    1.17D-38   1.0D+4
00083 C  VAX G-Format (D.P.)    1.0E-4    2.22D-308  1.0D+4

```

```

00084 C
00085 C*****
00086 C*****
00087 C
00088 C Error returns
00089 C
00090 C In case of an error, NCALC .NE. NB, and not all J's are
00091 C calculated to the desired accuracy.
00092 C
00093 C NCALC .LT. 0: An argument is out of range. For example,
00094 C NBES .LE. 0, ALPHA .LT. 0 or .GT. 1, or X is too large.
00095 C In this case, B(1) is set to zero, the remainder of the
00096 C B-vector is not calculated, and NCALC is set to
00097 C MIN(NB,0)-1 so that NCALC .NE. NB.
00098 C
00099 C NB .GT. NCALC .GT. 0: Not all requested function values could
00100 C be calculated accurately. This usually occurs because NB is
00101 C much larger than ABS(X). In this case, B(N) is calculated
00102 C to the desired accuracy for N .LE. NCALC, but precision
00103 C is lost for NCALC .LT. N .LE. NB. If B(N) does not vanish
00104 C for N .GT. NCALC (because it is too small to be represented),
00105 C and B(N)/B(NCALC) = 10**(-K), then only the first NSIG-K
00106 C significant figures of B(N) can be trusted.
00107 C
00108 C
00109 C Intrinsic and other functions required are:
00110 C
00111 C ABS, AINT, COS, DBLE, GAMMA (or DGAMMA), INT, MAX, MIN,
00112 C
00113 C REAL, SIN, SQRT
00114 C
00115 C
00116 C Acknowledgement
00117 C
00118 C This program is based on a program written by David J. Sookne
00119 C (2) that computes values of the Bessel functions J or I of real
00120 C argument and integer order. Modifications include the restriction
00121 C of the computation to the J Bessel function of non-negative real
00122 C argument, the extension of the computation to arbitrary positive
00123 C order, and the elimination of most underflow.
00124 C
00125 C References: "A Note on Backward Recurrence Algorithms," Olver,
00126 C F. W. J., and Sookne, D. J., Math. Comp. 26, 1972,
00127 C pp 941-947.
00128 C
00129 C "Bessel Functions of Real Argument and Integer Order,"
00130 C Sookne, D. J., NBS Jour. of Res. B. 77B, 1973, pp
00131 C 125-132.
00132 C
00133 C Latest modification: March 19, 1990
00134 C
00135 C Author: W. J. Cody
00136 C Applied Mathematics Division
00137 C Argonne National Laboratory
00138 C Argonne, IL 60439
00139 C
00140 C-----
00141 C INTEGER I,J,K,L,M,MAGX,N,NB,NBMX,NCALC,NEND,NSTART
00142 C REAL GAMMA,
00143 CD DOUBLE PRECISION DGAMMA,

```

```

00144      1 ALPHA,ALPEM,ALP2EM,B,CAPP,CAPQ,CONV,EIGHTH,EM,EN,ENMTEN,ENSIG,
00145      2 ENTEN,FACT,FOUR,FUNC,GNU,HALF,HALFX,ONE,ONE30,P,PI2,PLAST,
00146      3 POLD,PSAVE,PSAVEL,RTNSIG,S,SUM,T,T1,TEMPA,TEMPB,TEMPC,TEST,
00147      4 THREE,THREE5,TOVER,TWO,TWOFIV,TWOPI1,TWOPI2,X,XC,XIN,XK,XLARGE,
00148      5 XM,VCOS,VSIN,Z,ZERO
00149      DIMENSION B(NB), FACT(25)
00150  C-----
00151  C  Mathematical constants
00152  C
00153  C  PI2      - 2 / PI
00154  C  TWOPI1 - first few significant digits of 2 * PI
00155  C  TWOPI2 - (2*PI - TWOPI) to working precision, i.e.,
00156  C          TWOPI1 + TWOPI2 = 2 * PI to extra precision.
00157  C-----
00158      DATA PI2, TWOPI1, TWOPI2 /0.636619772367581343075535E0,6.28125E0,
00159      1 1.935307179586476925286767E-3/
00160      DATA ZERO, EIGHTH, HALF, ONE /0.0E0,0.125E0,0.5E0,1.0E0/
00161      DATA TWO, THREE, FOUR, TWOFIV /2.0E0,3.0E0,4.0E0,25.0E0/
00162      DATA ONE30, THREE5 /130.0E0,35.0E0/
00163  CD  DATA PI2, TWOPI1, TWOPI2 /0.636619772367581343075535D0,6.28125D0,
00164  CD  1 1.935307179586476925286767D-3/
00165  CD  DATA ZERO, EIGHTH, HALF, ONE /0.0D0,0.125D0,0.5D0,1.0D0/
00166  CD  DATA TWO, THREE, FOUR, TWOFIV /2.0D0,3.0D0,4.0D0,25.0D0/
00167  CD  DATA ONE30, THREE5 /130.0D0,35.0D0/
00168  C-----
00169  C  Machine-dependent parameters
00170  C-----
00171      DATA ENTEN, ENSIG, RTNSIG /1.0E38,1.0E8,1.0E-2/
00172      DATA ENMTEN, XLARGE /1.2E-37,1.0E4/
00173  CD  DATA ENTEN, ENSIG, RTNSIG /1.0D38,1.0D17,1.0D-4/
00174  CD  DATA ENMTEN, XLARGE /1.2D-37,1.0D4/
00175  C-----
00176  C  Factorial(N)
00177  C-----
00178      DATA FACT /1.0E0,1.0E0,2.0E0,6.0E0,24.0E0,1.2E2,7.2E2,5.04E3,
00179      1 4.032E4,3.6288E5,3.6288E6,3.99168E7,4.790016E8,6.2270208E9,
00180      2 8.71782912E10,1.307674368E12,2.0922789888E13,3.55687428096E14,
00181      3 6.402373705728E15,1.21645100408832E17,2.43290200817664E18,
00182      4 5.109094217170944E19,1.12400072777760768E21,
00183      5 2.585201673888497664E22,6.2044840173323943936E23/
00184  CD  DATA FACT /1.0D0,1.0D0,2.0D0,6.0D0,24.0D0,1.2D2,7.2D2,5.04D3,
00185  CD  1 4.032D4,3.6288D5,3.6288D6,3.99168D7,4.790016D8,6.2270208D9,
00186  CD  2 8.71782912D10,1.307674368D12,2.0922789888D13,3.55687428096D14,
00187  CD  3 6.402373705728D15,1.21645100408832D17,2.43290200817664D18,
00188  CD  4 5.109094217170944D19,1.12400072777760768D21,
00189  CD  5 2.585201673888497664D22,6.2044840173323943936D23/
00190  C-----
00191  C  Statement functions for conversion and the gamma function.
00192  C-----
00193      CONV(I) = REAL(I)
00194      FUNC(X) = GAMMA(X)
00195  CD  CONV(I) = DBLE(I)
00196  CD  FUNC(X) = DGAMMA(X)
00197  C-----
00198  C  Check for out of range arguments.
00199  C-----
00200      MAGX = INT(X)
00201      IF ((NB.GT.0) .AND. (X.GE.ZERO) .AND. (X.LE.XLARGE)
00202      1      .AND. (ALPHA.GE.ZERO) .AND. (ALPHA.LT.ONE))
00203      2      THEN

```

```

00204 C-----
00205 C Initialize result array to zero.
00206 C-----
00207         NCALC = NB
00208         DO 20 I=1,NB
00209             B(I) = ZERO
00210     20     CONTINUE
00211 C-----
00212 C Branch to use 2-term ascending series for small X and asymptotic
00213 C form for large X when NB is not too large.
00214 C-----
00215         IF (X.LT.RTNSIG) THEN
00216 C-----
00217 C Two-term ascending series for small X.
00218 C-----
00219         TEMPA = ONE
00220         ALPEM = ONE + ALPHA
00221         HALFX = ZERO
00222         IF (X.GT.ENMTEN) HALFX = HALF*X
00223         IF (ALPHA.NE.ZERO)
00224     1         TEMPA = HALFX**ALPHA/(ALPHA*FUNC(ALPHA))
00225         TEMPB = ZERO
00226         IF ((X+ONE).GT.ONE) TEMPB = -HALFX*HALFX
00227         B(1) = TEMPA + TEMPA*TEMPB/ALPEM
00228         IF ((X.NE.ZERO) .AND. (B(1).EQ.ZERO)) NCALC = 0
00229         IF (NB .NE. 1) THEN
00230             IF (X .LE. ZERO) THEN
00231                 DO 30 N=2,NB
00232                     B(N) = ZERO
00233     30     CONTINUE
00234             ELSE
00235 C-----
00236 C Calculate higher order functions.
00237 C-----
00238                 TEMPC = HALFX
00239                 TOVER = (ENMTEN+ENMTEN)/X
00240                 IF (TEMPB.NE.ZERO) TOVER = ENMTEN/TEMPB
00241                 DO 50 N=2,NB
00242                     TEMPA = TEMPA/ALPEM
00243                     ALPEM = ALPEM + ONE
00244                     TEMPA = TEMPA*TEMPC
00245                     IF (TEMPA.LE.TOVER*ALPEM) TEMPA = ZERO
00246                     B(N) = TEMPA + TEMPA*TEMPB/ALPEM
00247                     IF ((B(N).EQ.ZERO) .AND. (NCALC.GT.N))
00248     1                     NCALC = N-1
00249     50     CONTINUE
00250                 END IF
00251             END IF
00252             ELSE IF ((X.GT.TWOFIV) .AND. (NB.LE.MAGX+1)) THEN
00253 C-----
00254 C Asymptotic series for X .GT. 21.0.
00255 C-----
00256                 XC = SQRT(PI2/X)
00257                 XIN = (EIGHTH/X)**2
00258                 M = 11
00259                 IF (X.GE.THREE5) M = 8
00260                 IF (X.GE.ONE30) M = 4
00261                 XM = FOUR*CONV(M)
00262 C-----
00263 C Argument reduction for SIN and COS routines.

```

```

00264 C-----
00265 T = AINT(X/(TWOPI1+TWOPI2)+HALF)
00266 Z = ((X-T*TWOPI1)-T*TWOPI2) - (ALPHA+HALF)/PI2
00267 VSIN = SIN(Z)
00268 VCOS = COS(Z)
00269 GNU = ALPHA + ALPHA
00270 DO 80 I=1,2
00271 S = ((XM-ONE)-GNU)*((XM-ONE)+GNU)*XIN*HALF
00272 T = (GNU-(XM-THREE))*(GNU+(XM-THREE))
00273 CAPP = S*T/FACT(2*M+1)
00274 T1 = (GNU-(XM+ONE))*(GNU+(XM+ONE))
00275 CAPQ = S*T1/FACT(2*M+2)
00276 XK = XM
00277 K = M + M
00278 T1 = T
00279 DO 70 J=2,M
00280 XK = XK - FOUR
00281 S = ((XK-ONE)-GNU)*((XK-ONE)+GNU)
00282 T = (GNU-(XK-THREE))*(GNU+(XK-THREE))
00283 CAPP = (CAPP+ONE/FACT(K-1))*S*T*XIN
00284 CAPQ = (CAPQ+ONE/FACT(K))*S*T1*XIN
00285 K = K - 2
00286 T1 = T
00287 70 CONTINUE
00288 CAPP = CAPP + ONE
00289 CAPQ = (CAPQ+ONE)*(GNU*GNU-ONE)*(EIGHTH/X)
00290 B(I) = XC*(CAPP*VCOS-CAPQ*VSIN)
00291 IF (NB.EQ.1) GO TO 300
00292 T = VSIN
00293 VSIN = -VCOS
00294 VCOS = T
00295 GNU = GNU + TWO
00296 80 CONTINUE
00297 C-----
00298 C If NB .GT. 2, compute J(X,ORDER+I) I = 2, NB-1
00299 C-----
00300 IF (NB .GT. 2) THEN
00301 GNU = ALPHA + ALPHA + TWO
00302 DO 90 J=3,NB
00303 B(J) = GNU*B(J-1)/X - B(J-2)
00304 GNU = GNU + TWO
00305 90 CONTINUE
00306 END IF
00307 C-----
00308 C Use recurrence to generate results. First initialize the
00309 C calculation of P*S.
00310 C-----
00311 ELSE
00312 NBMX = NB - MAGX
00313 N = MAGX + 1
00314 EN = CONV(N+N) + (ALPHA+ALPHA)
00315 PLAST = ONE
00316 P = EN/X
00317 C-----
00318 C Calculate general significance test.
00319 C-----
00320 TEST = ENSIG + ENSIG
00321 IF (NBMX .GE. 3) THEN
00322 C-----
00323 C Calculate P*S until N = NB-1. Check for possible overflow.

```

```

00324 C-----
00325         TOVER = ENTEN/ENSIG
00326         NSTART = MAGX + 2
00327         NEND = NB - 1
00328         EN = CONV(NSTART+NSTART) - TWO + (ALPHA+ALPHA)
00329         DO 130 K=NSTART,NEND
00330             N = K
00331             EN = EN + TWO
00332             POLD = PLAST
00333             PLAST = P
00334             P = EN*PLAST/X - POLD
00335             IF (P.GT.TOVER) THEN
00336 C-----
00337 C To avoid overflow, divide P*S by TOVER. Calculate P*S until
00338 C ABS(P) .GT. 1.
00339 C-----
00340             TOVER = ENTEN
00341             P = P/TOVER
00342             PLAST = PLAST/TOVER
00343             PSAVE = P
00344             PSAVEL = PLAST
00345             NSTART = N + 1
00346 100         N = N + 1
00347             EN = EN + TWO
00348             POLD = PLAST
00349             PLAST = P
00350             P = EN*PLAST/X - POLD
00351             IF (P.LE.ONE) GO TO 100
00352             TEMPB = EN/X
00353 C-----
00354 C Calculate backward test and find NCALC, the highest N such that
00355 C the test is passed.
00356 C-----
00357             TEST = POLD*PLAST*(HALF-HALF/(TEMPB*TEMPB))
00358             TEST = TEST/ENSIG
00359             P = PLAST*TOVER
00360             N = N - 1
00361             EN = EN - TWO
00362             NEND = MIN(NB,N)
00363             DO 110 L=NSTART,NEND
00364                 POLD = PSAVEL
00365                 PSAVEL = PSAVE
00366                 PSAVE = EN*PSAVEL/X - POLD
00367                 IF (PSAVE*PSAVEL.GT.TEST) THEN
00368                     NCALC = L - 1
00369                     GO TO 190
00370                 END IF
00371 110         CONTINUE
00372             NCALC = NEND
00373             GO TO 190
00374         END IF
00375 130         CONTINUE
00376         N = NEND
00377         EN = CONV(N+N) + (ALPHA+ALPHA)
00378 C-----
00379 C Calculate special significance test for NBMX .GT. 2.
00380 C-----
00381             TEST = MAX(TEST,SQRT(PLAST*ENSIG)*SQRT(P+P))
00382             END IF
00383 C-----

```

```

00384 C Calculate P*S until significance test passes.
00385 C-----
00386     140          N = N + 1
00387             EN = EN + TWO
00388             POLD = PLAST
00389             PLAST = P
00390             P = EN*PLAST/X - POLD
00391             IF (P.LT.TEST) GO TO 140
00392 C-----
00393 C Initialize the backward recursion and the normalization sum.
00394 C-----
00395     190          N = N + 1
00396             EN = EN + TWO
00397             TEMPB = ZERO
00398             TEMPA = ONE/P
00399             M = 2*N - 4*(N/2)
00400             SUM = ZERO
00401             EM = CONV(N/2)
00402             ALPEM = (EM-ONE) + ALPHA
00403             ALP2EM = (EM+EM) + ALPHA
00404             IF (M .NE. 0) SUM = TEMPA*ALPEM*ALP2EM/EM
00405             NEND = N - NB
00406             IF (NEND .GT. 0) THEN
00407 C-----
00408 C Recur backward via difference equation, calculating (but not
00409 C storing) B(N), until N = NB.
00410 C-----
00411             DO 200 L=1,NEND
00412                 N = N - 1
00413                 EN = EN - TWO
00414                 TEMPC = TEMPB
00415                 TEMPB = TEMPA
00416                 TEMPA = (EN*TEMPB)/X - TEMPC
00417                 M = 2 - M
00418                 IF (M .NE. 0) THEN
00419                     EM = EM - ONE
00420                     ALP2EM = (EM+EM) + ALPHA
00421                     IF (N.EQ.1) GO TO 210
00422                     ALPEM = (EM-ONE) + ALPHA
00423                     IF (ALPEM.EQ.ZERO) ALPEM = ONE
00424                     SUM = (SUM+TEMPA*ALP2EM)*ALPEM/EM
00425                 END IF
00426     200          CONTINUE
00427             END IF
00428 C-----
00429 C Store B(NB).
00430 C-----
00431     210          B(N) = TEMPA
00432             IF (NEND .GE. 0) THEN
00433                 IF (NB .LE. 1) THEN
00434                     ALP2EM = ALPHA
00435                     IF ((ALPHA+ONE).EQ.ONE) ALP2EM = ONE
00436                     SUM = SUM + B(1)*ALP2EM
00437                     GO TO 250
00438                 ELSE
00439 C-----
00440 C Calculate and store B(NB-1).
00441 C-----
00442                 N = N - 1
00443                 EN = EN - TWO

```

```

00444          B(N) = (EN*TEMPA)/X - TEMPB
00445          IF (N.EQ.1) GO TO 240
00446          M = 2 - M
00447          IF (M .NE. 0) THEN
00448              EM = EM - ONE
00449              ALP2EM = (EM+EM) + ALPHA
00450              ALPEM = (EM-ONE) + ALPHA
00451              IF (ALPEM.EQ.ZERO) ALPEM = ONE
00452              SUM = (SUM+B(N)*ALP2EM)*ALPEM/EM
00453          END IF
00454      END IF
00455  END IF
00456  NEND = N - 2
00457  IF (NEND .NE. 0) THEN
00458      C-----
00459      C Calculate via difference equation and store B(N), until N = 2.
00460      C-----
00461          DO 230 L=1,NEND
00462              N = N - 1
00463              EN = EN - TWO
00464              B(N) = (EN*B(N+1))/X - B(N+2)
00465              M = 2 - M
00466              IF (M .NE. 0) THEN
00467                  EM = EM - ONE
00468                  ALP2EM = (EM+EM) + ALPHA
00469                  ALPEM = (EM-ONE) + ALPHA
00470                  IF (ALPEM.EQ.ZERO) ALPEM = ONE
00471                  SUM = (SUM+B(N)*ALP2EM)*ALPEM/EM
00472              END IF
00473          230      CONTINUE
00474      END IF
00475      C-----
00476      C Calculate B(1).
00477      C-----
00478          B(1) = TWO*(ALPHA+ONE)*B(2)/X - B(3)
00479          240      EM = EM - ONE
00480                  ALP2EM = (EM+EM) + ALPHA
00481                  IF (ALP2EM.EQ.ZERO) ALP2EM = ONE
00482                  SUM = SUM + B(1)*ALP2EM
00483      C-----
00484      C Normalize. Divide all B(N) by sum.
00485      C-----
00486          250      IF ((ALPHA+ONE).NE.ONE)
00487              1      SUM = SUM*FUNC(ALPHA)*(X*HALF)**(-ALPHA)
00488                  TEMPA = ENMTEN
00489                  IF (SUM.GT.ONE) TEMPA = TEMPA*SUM
00490                  DO 260 N=1,NB
00491                      IF (ABS(B(N)).LT.TEMPA) B(N) = ZERO
00492                      B(N) = B(N)/SUM
00493          260      CONTINUE
00494      END IF
00495      C-----
00496      C Error return -- X, NB, or ALPHA is out of range.
00497      C-----
00498      ELSE
00499          B(1) = ZERO
00500          NCALC = MIN(NB,0) - 1
00501      END IF
00502      C-----
00503      C Exit

```

```

00504 C-----
00505     300 RETURN
00506 C ----- Last line of RJBESL -----
00507     END

00001
00002
00003
00004
00005
00006     SUBROUTINE RYBESL(X,ALPHA,NB,BY,NCALC)
00007 C-----
00008 C
00009 C   This routine calculates Bessel functions Y SUB(N+ALPHA) (X)
00010 C   for non-negative argument X, and non-negative order N+ALPHA.
00011 C
00012 C
00013 C Explanation of variables in the calling sequence
00014 C
00015 C X      - Working precision non-negative real argument for which
00016 C          Y's are to be calculated.
00017 C ALPHA  - Working precision fractional part of order for which
00018 C          Y's are to be calculated.  0 .LE. ALPHA .LT. 1.0.
00019 C NB     - Integer number of functions to be calculated, NB .GT. 0.
00020 C          The first function calculated is of order ALPHA, and the
00021 C          last is of order (NB - 1 + ALPHA).
00022 C BY     - Working precision output vector of length NB.  If the
00023 C          routine terminates normally (NCALC=NB), the vector BY
00024 C          contains the functions Y(ALPHA,X), ... , Y(NB-1+ALPHA,X),
00025 C          If (0 .LT. NCALC .LT. NB), BY(I) contains correct function
00026 C          values for I .LE. NCALC, and contains the ratios
00027 C          Y(ALPHA+I-1,X)/Y(ALPHA+I-2,X) for the rest of the array.
00028 C NCALC  - Integer output variable indicating possible errors.
00029 C          Before using the vector BY, the user should check that
00030 C          NCALC=NB, i.e., all orders have been calculated to
00031 C          the desired accuracy.  See error returns below.
00032 C
00033 C
00034 C*****
00035 C*****
00036 C
00037 C Explanation of machine-dependent constants
00038 C
00039 C   beta   = Radix for the floating-point system
00040 C   p      = Number of significant base-beta digits in the
00041 C            significand of a floating-point number
00042 C   minexp  = Smallest representable power of beta
00043 C   maxexp  = Smallest power of beta that overflows
00044 C   EPS     = beta ** (-p)
00045 C   DEL     = Machine number below which sin(x)/x = 1; approximately
00046 C            SQRT(EPS).
00047 C   XMIN    = Smallest acceptable argument for RBESY; approximately
00048 C            max(2*beta**minexp,2/XINF), rounded up
00049 C   XINF    = Largest positive machine number; approximately
00050 C            beta**maxexp
00051 C   THRESH  = Lower bound for use of the asymptotic form; approximately
00052 C            AINT(-LOG10(EPS/2.0))+1.0
00053 C   XLARGE  = Upper bound on X; approximately 1/DEL, because the sine
00054 C            and cosine functions have lost about half of their
00055 C            precision at that point.

```

```

00056 C
00057 C
00058 C     Approximate values for some important machines are:
00059 C
00060 C             beta      p      minexp      maxexp      EPS
00061 C
00062 C CRAY-1      (S.P.)    2      48      -8193      8191      3.55E-15
00063 C Cyber 180/185
00064 C   under NOS  (S.P.)    2      48      -975      1070      3.55E-15
00065 C IEEE (IBM/XT,
00066 C   SUN, etc.) (S.P.)    2      24      -126      128      5.96E-8
00067 C IEEE (IBM/XT,
00068 C   SUN, etc.) (D.P.)    2      53     -1022     1024     1.11D-16
00069 C IBM 3033     (D.P.)   16      14       -65        63     1.39D-17
00070 C VAX          (S.P.)    2      24      -128      127      5.96E-8
00071 C VAX D-Format (D.P.)    2      56      -128      127     1.39D-17
00072 C VAX G-Format (D.P.)    2      53     -1024     1023     1.11D-16
00073 C
00074 C
00075 C             DEL      XMIN      XINF      THRESH  XLARGE
00076 C
00077 C CRAY-1      (S.P.)   5.0E-8  3.67E-2466  5.45E+2465  15.0E0  2.0E7
00078 C Cyber 180/855
00079 C   under NOS  (S.P.)   5.0E-8  6.28E-294  1.26E+322  15.0E0  2.0E7
00080 C IEEE (IBM/XT,
00081 C   SUN, etc.) (S.P.)   1.0E-4  2.36E-38  3.40E+38   8.0E0  1.0E4
00082 C IEEE (IBM/XT,
00083 C   SUN, etc.) (D.P.)   1.0D-8  4.46D-308  1.79D+308  16.0D0  1.0D8
00084 C IBM 3033     (D.P.)   1.0D-8  2.77D-76  7.23D+75  17.0D0  1.0D8
00085 C VAX          (S.P.)   1.0E-4  1.18E-38  1.70E+38   8.0E0  1.0E4
00086 C VAX D-Format (D.P.)   1.0D-9  1.18D-38  1.70D+38  17.0D0  1.0D9
00087 C VAX G-Format (D.P.)   1.0D-8  2.23D-308  8.98D+307  16.0D0  1.0D8
00088 C
00089 C*****
00090 C*****
00091 C
00092 C Error returns
00093 C
00094 C In case of an error, NCALC .NE. NB, and not all Y's are
00095 C calculated to the desired accuracy.
00096 C
00097 C NCALC .LT. -1: An argument is out of range. For example,
00098 C   NB .LE. 0, IZE is not 1 or 2, or IZE=1 and ABS(X) .GE.
00099 C   XMAX. In this case, BY(1) = 0.0, the remainder of the
00100 C   BY-vector is not calculated, and NCALC is set to
00101 C   MIN0(NB,0)-2 so that NCALC .NE. NB.
00102 C NCALC = -1: Y(ALPHA,X) .GE. XINF. The requested function
00103 C   values are set to 0.0.
00104 C 1 .LT. NCALC .LT. NB: Not all requested function values could
00105 C   be calculated accurately. BY(I) contains correct function
00106 C   values for I .LE. NCALC, and and the remaining NB-NCALC
00107 C   array elements contain 0.0.
00108 C
00109 C
00110 C Intrinsic functions required are:
00111 C
00112 C   DBLE, EXP, INT, MAX, MIN, REAL, SQRT
00113 C
00114 C
00115 C Acknowledgement

```

```

00116 C
00117 C This program draws heavily on Temme's Algol program for Y(a,x)
00118 C and Y(a+1,x) and on Campbell's programs for Y_nu(x). Temme's
00119 C scheme is used for x < THRESH, and Campbell's scheme is used
00120 C in the asymptotic region. Segments of code from both sources
00121 C have been translated into Fortran 77, merged, and heavily modified.
00122 C Modifications include parameterization of machine dependencies,
00123 C use of a new approximation for ln(gamma(x)), and built-in
00124 C protection against over/underflow.
00125 C
00126 C References: "Bessel functions J_nu(x) and Y_nu(x) of real
00127 C order and real argument," Campbell, J. B.,
00128 C Comp. Phy. Comm. 18, 1979, pp. 133-142.
00129 C
00130 C "On the numerical evaluation of the ordinary
00131 C Bessel function of the second kind," Temme,
00132 C N. M., J. Comput. Phys. 21, 1976, pp. 343-350.
00133 C
00134 C Latest modification: March 19, 1990
00135 C
00136 C Modified by: W. J. Cody
00137 C Applied Mathematics Division
00138 C Argonne National Laboratory
00139 C Argonne, IL 60439
00140 C
00141 C-----
00142 C INTEGER I,K,NA,NB,NCALC
00143 C REAL
00144 CD DOUBLE PRECISION
00145 1 ALFA,ALPHA,AYE,B,BY,C,CH,COSMU,D,DEL,DEN,DDIV,DIV,DMU,D1,D2,
00146 2 E,EIGHT,EN,ENU,EN1,EPS,EVEN,EX,F,FIVPI,G,GAMMA,H,HALF,ODD,
00147 3 ONBPI,ONE,ONE5,P,PA,PA1,PI,PIBY2,PIM5,Q,QA,QA1,Q0,R,S,SINMU,
00148 4 SQ2BPI,TEN9,TERM,THREE,THRESH,TWO,TWOBYX,X,XINF,XLARGE,XMIN,
00149 5 XNA,X2,YA,YA1,ZERO
00150 C DIMENSION BY(NB),CH(21)
00151 C-----
00152 C Mathematical constants
00153 C FIVPI = 5*PI
00154 C PIM5 = 5*PI - 15
00155 C ONBPI = 1/PI
00156 C PIBY2 = PI/2
00157 C SQ2BPI = SQUARE ROOT OF 2/PI
00158 C-----
00159 C DATA ZERO,HALF,ONE,TWO,THREE/0.0E0,0.5E0,1.0E0,2.0E0,3.0E0/
00160 C DATA EIGHT,ONE5,TEN9/8.0E0,15.0E0,1.9E1/
00161 C DATA FIVPI,PIBY2/1.5707963267948966192E1,1.5707963267948966192E0/
00162 C DATA PI,SQ2BPI/3.1415926535897932385E0,7.9788456080286535588E-1/
00163 C DATA PIM5,ONBPI/7.0796326794896619231E-1,3.1830988618379067154E-1/
00164 CD DATA ZERO,HALF,ONE,TWO,THREE/0.0D0,0.5D0,1.0D0,2.0D0,3.0D0/
00165 CD DATA EIGHT,ONE5,TEN9/8.0D0,15.0D0,1.9D1/
00166 CD DATA FIVPI,PIBY2/1.5707963267948966192D1,1.5707963267948966192D0/
00167 CD DATA PI,SQ2BPI/3.1415926535897932385D0,7.9788456080286535588D-1/
00168 CD DATA PIM5,ONBPI/7.0796326794896619231D-1,3.1830988618379067154D-1/
00169 C-----
00170 C Machine-dependent constants
00171 C-----
00172 C DATA DEL,XMIN,XINF,EPS/1.0E-4,2.36E-38,1.70E38,5.96E-8/
00173 C DATA THRESH,XLARGE/8.0E0,1.0E4/
00174 CD DATA DEL,XMIN,XINF,EPS/1.0D-8,4.46D-308,1.79D308,1.11D-16/
00175 CD DATA THRESH,XLARGE/16.0D0,1.0D8/

```

```

00176 C-----
00177 C Coefficients for Chebyshev polynomial expansion of
00178 C      1/gamma(1-x), abs(x) .le. .5
00179 C-----
00180 DATA CH/-0.67735241822398840964E-23,-0.61455180116049879894E-22,
00181 1      0.29017595056104745456E-20, 0.13639417919073099464E-18,
00182 2      0.23826220476859635824E-17,-0.90642907957550702534E-17,
00183 3      -0.14943667065169001769E-14,-0.33919078305362211264E-13,
00184 4      -0.17023776642512729175E-12, 0.91609750938768647911E-11,
00185 5      0.24230957900482704055E-09, 0.17451364971382984243E-08,
00186 6      -0.33126119768180852711E-07,-0.86592079961391259661E-06,
00187 7      -0.49717367041957398581E-05, 0.76309597585908126618E-04,
00188 8      0.12719271366545622927E-02, 0.17063050710955562222E-02,
00189 9      -0.76852840844786673690E-01,-0.28387654227602353814E+00,
00190 A      0.92187029365045265648E+00/
00191 CD DATA CH/-0.67735241822398840964D-23,-0.61455180116049879894D-22,
00192 CD 1      0.29017595056104745456D-20, 0.13639417919073099464D-18,
00193 CD 2      0.23826220476859635824D-17,-0.90642907957550702534D-17,
00194 CD 3      -0.14943667065169001769D-14,-0.33919078305362211264D-13,
00195 CD 4      -0.17023776642512729175D-12, 0.91609750938768647911D-11,
00196 CD 5      0.24230957900482704055D-09, 0.17451364971382984243D-08,
00197 CD 6      -0.33126119768180852711D-07,-0.86592079961391259661D-06,
00198 CD 7      -0.49717367041957398581D-05, 0.76309597585908126618D-04,
00199 CD 8      0.12719271366545622927D-02, 0.17063050710955562222D-02,
00200 CD 9      -0.76852840844786673690D-01,-0.28387654227602353814D+00,
00201 CD A      0.92187029365045265648D+00/
00202 C-----
00203 EX = X
00204 ENU = ALPHA
00205 IF ((NB .GT. 0) .AND. (X .GE. XMIN) .AND. (EX .LT. XLARGE)
00206 1      .AND. (ENU .GE. ZERO) .AND. (ENU .LT. ONE)) THEN
00207 XNA = AINT(ENU+HALF)
00208 NA = INT(XNA)
00209 IF (NA .EQ. 1) ENU = ENU - XNA
00210 IF (ENU .EQ. -HALF) THEN
00211 P = SQ2BPI/SQRT(EX)
00212 YA = P * SIN(EX)
00213 YA1 = -P * COS(EX)
00214 ELSE IF (EX .LT. THREE) THEN
00215 C-----
00216 C Use Temme's scheme for small X
00217 C-----
00218 B = EX * HALF
00219 D = -LOG(B)
00220 F = ENU * D
00221 E = B**(-ENU)
00222 IF (ABS(ENU) .LT. DEL) THEN
00223 C = ONBPI
00224 ELSE
00225 C = ENU / SIN(ENU*PI)
00226 END IF
00227 C-----
00228 C Computation of sinh(f)/f
00229 C-----
00230 IF (ABS(F) .LT. ONE) THEN
00231 X2 = F*F
00232 EN = TEN9
00233 S = ONE
00234 DO 80 I = 1, 9
00235 S = S*X2/EN/(EN-ONE)+ONE

```

```

00236             EN = EN - TWO
00237      80             CONTINUE
00238             ELSE
00239                 S = (E - ONE/E) * HALF / F
00240             END IF
00241 C-----
00242 C  Computation of 1/gamma(1-a) using Chebyshev polynomials
00243 C-----
00244             X2 = ENU*ENU*EIGHT
00245             AYE = CH(1)
00246             EVEN = ZERO
00247             ALFA = CH(2)
00248             ODD = ZERO
00249             DO 40 I = 3, 19, 2
00250                 EVEN = -(AYE+AYE+EVEN)
00251                 AYE = -EVEN*X2 - AYE + CH(I)
00252                 ODD = -(ALFA+ALFA+ODD)
00253                 ALFA = -ODD*X2 - ALFA + CH(I+1)
00254      40             CONTINUE
00255             EVEN = (EVEN*HALF+AYE)*X2 - AYE + CH(21)
00256             ODD = (ODD+ALFA)*TWO
00257             GAMMA = ODD*ENU + EVEN
00258 C-----
00259 C  End of computation of 1/gamma(1-a)
00260 C-----
00261             G = E * GAMMA
00262             E = (E + ONE/E) * HALF
00263             F = TWO*C*(ODD*E+EVEN*S*D)
00264             E = ENU*ENU
00265             P = G*C
00266             Q = ONBPI / G
00267             C = ENU*PIBY2
00268             IF (ABS(C) .LT. DEL) THEN
00269                 R = ONE
00270             ELSE
00271                 R = SIN(C)/C
00272             END IF
00273             R = PI*C*R*R
00274             C = ONE
00275             D = - B*B
00276             H = ZERO
00277             YA = F + R*Q
00278             YA1 = P
00279             EN = ZERO
00280      100            EN = EN + ONE
00281             IF (ABS(G/(ONE+ABS(YA)))
00282      1             + ABS(H/(ONE+ABS(YA1)))) .GT. EPS) THEN
00283                 F = (F*EN+P+Q)/(EN*EN-E)
00284                 C = C * D/EN
00285                 P = P/(EN-ENU)
00286                 Q = Q/(EN+ENU)
00287                 G = C*(F+R*Q)
00288                 H = C*P - EN*G
00289                 YA = YA + G
00290                 YA1 = YA1+H
00291                 GO TO 100
00292             END IF
00293             YA = -YA
00294             YA1 = -YA1/B
00295             ELSE IF (EX .LT. THRESH) THEN

```

```

00296 C-----
00297 C  Use Temme's scheme for moderate X
00298 C-----
00299          C = (HALF-ENU)*(HALF+ENU)
00300          B = EX + EX
00301          E = (EX*ONBPI*COS(ENU*PI)/EPS)
00302          E = E*E
00303          P = ONE
00304          Q = -EX
00305          R = ONE + EX*EX
00306          S = R
00307          EN = TWO
00308      200      IF (R*EN*EN .LT. E) THEN
00309                  EN1 = EN+ONE
00310                  D = (EN-ONE+C/EN)/S
00311                  P = (EN+EN-P*D)/EN1
00312                  Q = (-B+Q*D)/EN1
00313                  S = P*P + Q*Q
00314                  R = R*S
00315                  EN = EN1
00316                  GO TO 200
00317      END IF
00318      F = P/S
00319      P = F
00320      G = -Q/S
00321      Q = G
00322      220      EN = EN - ONE
00323      IF (EN .GT. ZERO) THEN
00324          R = EN1*(TWO-P)-TWO
00325          S = B + EN1*Q
00326          D = (EN-ONE+C/EN)/(R*R+S*S)
00327          P = D*R
00328          Q = D*S
00329          E = F + ONE
00330          F = P*E - G*Q
00331          G = Q*E + P*G
00332          EN1 = EN
00333          GO TO 220
00334      END IF
00335      F = ONE + F
00336      D = F*F + G*G
00337      PA = F/D
00338      QA = -G/D
00339      D = ENU + HALF -P
00340      Q = Q + EX
00341      PA1 = (PA*Q-QA*D)/EX
00342      QA1 = (QA*Q+PA*D)/EX
00343      B = EX - PIBY2*(ENU+HALF)
00344      C = COS(B)
00345      S = SIN(B)
00346      D = SQ2BPI/SQRT(EX)
00347      YA = D*(PA*S+QA*C)
00348      YA1 = D*(QA1*S-PA1*C)
00349      ELSE
00350 C-----
00351 C  Use Campbell's asymptotic scheme.
00352 C-----
00353          NA = 0
00354          D1 = AINT(EX/FIVPI)
00355          I = INT(D1)

```

```

00356          DMU = ((EX-ONE5*D1)-D1*PIM5)-(ALPHA+HALF)*PIBY2
00357          IF (I-2*(I/2) .EQ. 0) THEN
00358              COSMU = COS(DMU)
00359              SINMU = SIN(DMU)
00360          ELSE
00361              COSMU = -COS(DMU)
00362              SINMU = -SIN(DMU)
00363          END IF
00364          DDIV = EIGHT * EX
00365          DMU = ALPHA
00366          DEN = SQRT(EX)
00367          DO 350 K = 1, 2
00368              P = COSMU
00369              COSMU = SINMU
00370              SINMU = -P
00371              D1 = (TWO*DMU-ONE)*(TWO*DMU+ONE)
00372              D2 = ZERO
00373              DIV = DDIV
00374              P = ZERO
00375              Q = ZERO
00376              Q0 = D1/DIV
00377              TERM = Q0
00378              DO 310 I = 2, 20
00379                  D2 = D2 + EIGHT
00380                  D1 = D1 - D2
00381                  DIV = DIV + DDIV
00382                  TERM = -TERM*D1/DIV
00383                  P = P + TERM
00384                  D2 = D2 + EIGHT
00385                  D1 = D1 - D2
00386                  DIV = DIV + DDIV
00387                  TERM = TERM*D1/DIV
00388                  Q = Q + TERM
00389                  IF (ABS(TERM) .LE. EPS) GO TO 320
00390          310          CONTINUE
00391          320          P = P + ONE
00392                  Q = Q + Q0
00393                  IF (K .EQ. 1) THEN
00394                      YA = SQ2BPI * (P*COSMU-Q*SINMU) / DEN
00395                  ELSE
00396                      YA1 = SQ2BPI * (P*COSMU-Q*SINMU) / DEN
00397                  END IF
00398                  DMU = DMU + ONE
00399          350          CONTINUE
00400          END IF
00401          IF (NA .EQ. 1) THEN
00402              H = TWO*(ENU+ONE)/EX
00403              IF (H .GT. ONE) THEN
00404                  IF (ABS(YA1) .GT. XINF/H) THEN
00405                      H = ZERO
00406                      YA = ZERO
00407                  END IF
00408              END IF
00409              H = H*YA1 - YA
00410              YA = YA1
00411              YA1 = H
00412          END IF
00413          C-----
00414          C  Now have first one or two Y's
00415          C-----

```

```

00416          BY(1) = YA
00417          BY(2) = YA1
00418          IF (YA1 .EQ. ZERO) THEN
00419              NCALC = 1
00420          ELSE
00421              AYE = ONE + ALPHA
00422              TWOBYX = TWO/EX
00423              NCALC = 2
00424              DO 400 I = 3, NB
00425                  IF (TWOBYX .LT. ONE) THEN
00426                      IF (ABS(BY(I-1))*TWOBYX .GE. XINF/AYE)
00427                          1 GO TO 450
00428                      ELSE
00429                          IF (ABS(BY(I-1)) .GE. XINF/AYE/TWOBYX )
00430                          1 GO TO 450
00431                      END IF
00432                      BY(I) = TWOBYX*AYE*BY(I-1) - BY(I-2)
00433                      AYE = AYE + ONE
00434                      NCALC = NCALC + 1
00435              400 CONTINUE
00436          END IF
00437          450 DO 460 I = NCALC+1, NB
00438              BY(I) = ZERO
00439          460 CONTINUE
00440          ELSE
00441              BY(1) = ZERO
00442              NCALC = MIN(NB,0) - 1
00443          END IF
00444          900 RETURN
00445 C----- Last line of RYBESL -----
00446          END

```

```

00001          REAL FUNCTION GAMMA(X)
00002 CD          DOUBLE PRECISION FUNCTION DGAMMA(X)
00003 C-----
00004 C
00005 C This routine calculates the GAMMA function for a real argument X.
00006 C Computation is based on an algorithm outlined in reference 1.
00007 C The program uses rational functions that approximate the GAMMA
00008 C function to at least 20 significant decimal digits. Coefficients
00009 C for the approximation over the interval (1,2) are unpublished.
00010 C Those for the approximation for X .GE. 12 are from reference 2.
00011 C The accuracy achieved depends on the arithmetic system, the
00012 C compiler, the intrinsic functions, and proper selection of the
00013 C machine-dependent constants.
00014 C
00015 C
00016 C*****
00017 C*****
00018 C
00019 C Explanation of machine-dependent constants
00020 C
00021 C beta - radix for the floating-point representation
00022 C maxexp - the smallest positive power of beta that overflows
00023 C XBIG - the largest argument for which GAMMA(X) is representable
00024 C in the machine, i.e., the solution to the equation
00025 C GAMMA(XBIG) = beta**maxexp
00026 C XINF - the largest machine representable floating-point number;
00027 C approximately beta**maxexp
00028 C EPS - the smallest positive floating-point number such that

```

```

00029 C          1.0+EPS .GT. 1.0
00030 C XMININ - the smallest positive floating-point number such that
00031 C          1/XMININ is machine representable
00032 C
00033 C      Approximate values for some important machines are:
00034 C
00035 C                      beta          maxexp          XBIG
00036 C
00037 C CRAY-1          (S.P.)          2          8191          966.961
00038 C Cyber 180/855
00039 C   under NOS      (S.P.)          2          1070          177.803
00040 C IEEE (IBM/XT,
00041 C   SUN, etc.)      (S.P.)          2          128          35.040
00042 C IEEE (IBM/XT,
00043 C   SUN, etc.)      (D.P.)          2          1024          171.624
00044 C IBM 3033          (D.P.)          16          63          57.574
00045 C VAX D-Format      (D.P.)          2          127          34.844
00046 C VAX G-Format      (D.P.)          2          1023          171.489
00047 C
00048 C                      XINF          EPS          XMININ
00049 C
00050 C CRAY-1          (S.P.)    5.45E+2465    7.11E-15    1.84E-2466
00051 C Cyber 180/855
00052 C   under NOS      (S.P.)    1.26E+322    3.55E-15    3.14E-294
00053 C IEEE (IBM/XT,
00054 C   SUN, etc.)      (S.P.)    3.40E+38    1.19E-7     1.18E-38
00055 C IEEE (IBM/XT,
00056 C   SUN, etc.)      (D.P.)    1.79D+308    2.22D-16    2.23D-308
00057 C IBM 3033          (D.P.)    7.23D+75    2.22D-16    1.39D-76
00058 C VAX D-Format      (D.P.)    1.70D+38    1.39D-17    5.88D-39
00059 C VAX G-Format      (D.P.)    8.98D+307    1.11D-16    1.12D-308
00060 C
00061 C*****
00062 C*****
00063 C
00064 C Error returns
00065 C
00066 C The program returns the value XINF for singularities or
00067 C   when overflow would occur. The computation is believed
00068 C   to be free of underflow and overflow.
00069 C
00070 C
00071 C Intrinsic functions required are:
00072 C
00073 C   INT, DBLE, EXP, LOG, REAL, SIN
00074 C
00075 C
00076 C References: "An Overview of Software Development for Special
00077 C             Functions," W. J. Cody, Lecture Notes in Mathematics,
00078 C             506, Numerical Analysis Dundee, 1975, G. A. Watson
00079 C             (ed.), Springer Verlag, Berlin, 1976.
00080 C
00081 C             Computer Approximations, Hart, Et. Al., Wiley and
00082 C             sons, New York, 1968.
00083 C
00084 C Latest modification: October 12, 1989
00085 C
00086 C Authors: W. J. Cody and L. Stoltz
00087 C           Applied Mathematics Division
00088 C           Argonne National Laboratory

```

```

00089 C Argonne, IL 60439
00090 C
00091 C-----
00092 INTEGER I,N
00093 LOGICAL PARITY
00094 REAL
00095 CD DOUBLE PRECISION
00096 1 C, CONV, EPS, FACT, HALF, ONE, P, PI, Q, RES, SQRTPI, SUM, TWELVE,
00097 2 TWO, X, XBIG, XDEN, XINF, XMININ, XNUM, Y, Y1, YSQ, Z, ZERO
00098 DIMENSION C(7), P(8), Q(8)
00099 C-----
00100 C Mathematical constants
00101 C-----
00102 DATA ONE, HALF, TWELVE, TWO, ZERO/1.0E0, 0.5E0, 12.0E0, 2.0E0, 0.0E0/,
00103 1 SQRTPI/0.9189385332046727417803297E0/,
00104 2 PI/3.1415926535897932384626434E0/
00105 CD DATA ONE, HALF, TWELVE, TWO, ZERO/1.0D0, 0.5D0, 12.0D0, 2.0D0, 0.0D0/,
00106 CD 1 SQRTPI/0.9189385332046727417803297D0/,
00107 CD 2 PI/3.1415926535897932384626434D0/
00108 C-----
00109 C Machine dependent parameters
00110 C-----
00111 DATA XBIG, XMININ, EPS/35.040E0, 1.18E-38, 1.19E-7/,
00112 1 XINF/1.7E38/
00113 CD DATA XBIG, XMININ, EPS/171.624D0, 2.23D-308, 2.22D-16/,
00114 CD 1 XINF/1.79D308/
00115 C-----
00116 C Numerator and denominator coefficients for rational minimax
00117 C approximation over (1,2).
00118 C-----
00120 DATA P/-1.71618513886549492533811E+0, 2.47656508055759199108314E+1,
00121 1 -3.79804256470945635097577E+2, 6.29331155312818442661052E+2,
00122 2 8.66966202790413211295064E+2, -3.14512729688483675254357E+4,
00123 3 -3.61444134186911729807069E+4, 6.64561438202405440627855E+4/
00124 DATA Q/-3.08402300119738975254353E+1, 3.15350626979604161529144E+2,
00125 1 -1.01515636749021914166146E+3, -3.10777167157231109440444E+3,
00126 2 2.25381184209801510330112E+4, 4.75584627752788110767815E+3,
00127 3 -1.34659959864969306392456E+5, -1.15132259675553483497211E+5/
00128 CD DATA P/-1.71618513886549492533811D+0, 2.47656508055759199108314D+1,
00129 CD 1 -3.79804256470945635097577D+2, 6.29331155312818442661052D+2,
00130 CD 2 8.66966202790413211295064D+2, -3.14512729688483675254357D+4,
00131 CD 3 -3.61444134186911729807069D+4, 6.64561438202405440627855D+4/
00132 CD DATA Q/-3.08402300119738975254353D+1, 3.15350626979604161529144D+2,
00133 CD 1 -1.01515636749021914166146D+3, -3.10777167157231109440444D+3,
00134 CD 2 2.25381184209801510330112D+4, 4.75584627752788110767815D+3,
00135 CD 3 -1.34659959864969306392456D+5, -1.15132259675553483497211D+5/
00136 C-----
00137 C Coefficients for minimax approximation over (12, INF).
00138 C-----
00139 DATA C/-1.910444077728E-03, 8.4171387781295E-04,
00140 1 -5.952379913043012E-04, 7.93650793500350248E-04,
00141 2 -2.777777777777681622553E-03, 8.3333333333333331554247E-02,
00142 3 5.7083835261E-03/
00143 CD DATA C/-1.910444077728D-03, 8.4171387781295D-04,
00144 CD 1 -5.952379913043012D-04, 7.93650793500350248D-04,
00145 CD 2 -2.777777777777681622553D-03, 8.3333333333333331554247D-02,
00146 CD 3 5.7083835261D-03/
00147 C-----
00148 C Statement functions for conversion between integer and float
00149 C-----

```

```

00149      CONV(I) = REAL(I)
00150      CD      CONV(I) = DBLE(I)
00151      PARITY = .FALSE.
00152      FACT = ONE
00153      N = 0
00154      Y = X
00155      IF (Y .LE. ZERO) THEN
00156      C-----
00157      C  Argument is negative
00158      C-----
00159          Y = -X
00160          Y1 = AINT(Y)
00161          RES = Y - Y1
00162          IF (RES .NE. ZERO) THEN
00163              IF (Y1 .NE. AINT(Y1*HALF)*TWO) PARITY = .TRUE.
00164              FACT = -PI / SIN(PI*RES)
00165              Y = Y + ONE
00166          ELSE
00167              RES = XINF
00168              GO TO 900
00169          END IF
00170      END IF
00171      C-----
00172      C  Argument is positive
00173      C-----
00174          IF (Y .LT. EPS) THEN
00175      C-----
00176      C  Argument .LT. EPS
00177      C-----
00178          IF (Y .GE. XMININ) THEN
00179              RES = ONE / Y
00180          ELSE
00181              RES = XINF
00182              GO TO 900
00183          END IF
00184          ELSE IF (Y .LT. TWELVE) THEN
00185              Y1 = Y
00186              IF (Y .LT. ONE) THEN
00187      C-----
00188      C  0.0 .LT. argument .LT. 1.0
00189      C-----
00190                  Z = Y
00191                  Y = Y + ONE
00192              ELSE
00193      C-----
00194      C  1.0 .LT. argument .LT. 12.0, reduce argument if necessary
00195      C-----
00196                  N = INT(Y) - 1
00197                  Y = Y - CONV(N)
00198                  Z = Y - ONE
00199              END IF
00200      C-----
00201      C  Evaluate approximation for 1.0 .LT. argument .LT. 2.0
00202      C-----
00203          XNUM = ZERO
00204          XDEN = ONE
00205          DO 260 I = 1, 8
00206              XNUM = (XNUM + P(I)) * Z
00207              XDEN = XDEN * Z + Q(I)
00208      260      CONTINUE

```

```

00209             RES = XNUM / XDEN + ONE
00210             IF (Y1 .LT. Y) THEN
00211 C-----
00212 C   Adjust result for case  0.0 .LT. argument .LT. 1.0
00213 C-----
00214             RES = RES / Y1
00215             ELSE IF (Y1 .GT. Y) THEN
00216 C-----
00217 C   Adjust result for case  2.0 .LT. argument .LT. 12.0
00218 C-----
00219             DO 290 I = 1, N
00220                 RES = RES * Y
00221                 Y = Y + ONE
00222 290          CONTINUE
00223             END IF
00224             ELSE
00225 C-----
00226 C   Evaluate for argument .GE. 12.0,
00227 C-----
00228             IF (Y .LE. XBIG) THEN
00229                 YSQ = Y * Y
00230                 SUM = C(7)
00231                 DO 350 I = 1, 6
00232                     SUM = SUM / YSQ + C(I)
00233 350          CONTINUE
00234                 SUM = SUM/Y - Y + SQRTPI
00235                 SUM = SUM + (Y-HALF)*LOG(Y)
00236                 RES = EXP(SUM)
00237             ELSE
00238                 RES = XINF
00239                 GO TO 900
00240             END IF
00241         END IF
00242 C-----
00243 C   Final adjustments and return
00244 C-----
00245             IF (PARITY) RES = -RES
00246             IF (FACT .NE. ONE) RES = FACT / RES
00247 900      GAMMA = RES
00248 CD900    DGAMMA = RES
00249             RETURN
00250             END

00001 C ----- Last line of GAMMA -----
00002
00003     FUNCTION SINH(X)
00004     SINH  = (EXP(X)-EXP(-X))/2.
00005     RETURN
00006     END

00001     FUNCTION TANH(X)
00002     TX    = 2.*X
00003     ETX   = EXP(-TX)
00004     TANH  = (1.-ETX)/(1.+ETX)
00005     RETURN
00006     END

00001     FUNCTION ARCSECH(X)

```

```

00002      RX      =1./X
00003      TRM      = SQRT(RX**2-1.)
00004      ARCSECH = ALOG(RX+TRM)
00005      RETURN
00006      END

00001      FUNCTION COTH(X)
00002      COTH      = 1./TANH(X)
00003      RETURN
00004      END

00001      FUNCTION U1(X)
00002      X3        = X**3
00003      U1        = (3.*X-5.*X3)/24.
00004      RETURN
00005      END

00001      FUNCTION U2(X)
00002      X2        = X**2
00003      X4        = X2**2
00004      X6        = X2*X4
00005      U2        = (81.*X2-462.*X4+385.*X6)/1152.
00006      RETURN
00007      END

00001      FUNCTION V1(X)
00002      X3        = X**3
00003      V1        = (-9.*X+7.*X3)/24.
00004      RETURN
00005      END

00001      FUNCTION V2(X)
00002      X2        = X**2
00003      X4        = X2**2
00004      X6        = X2*X4
00005      V2        = (-135.*X2+594.*X4-455.*X6)/1152.
00006      RETURN
00007      END

00001      SUBROUTINE ABESJ ( ARG,ORD,BESJ )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESJ NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI        = 3.14159265
00005      ORD2      = ORD**2
00006      SECHAL   = ARG/ORD
00007      AL       = ARCSECH( SECHAL )
00008      TANHALL  = TANH( AL )
00009      COTHALL  = 1./TANHALL
00010      RNUMJ    = EXP( ORD*( TANHALL-AL ) )
00011      DENJ     = SQRT( 2.*PI*ORD*TANHALL )
00012      BESJ     = (RNUMJ/DENJ)*(1.+U1( COTHALL )/ORD+U2( COTHALL )/ORD2)
00013 100    CONTINUE
00014      RETURN
00015      END

```

```

00001      SUBROUTINE ABESY ( ARG,ORD,BESY,RBESY )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESY NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI      = 3.14159265
00005      ORD2     = ORD**2
00006      SECHAL  = ARG/ORD
00007      AL      = ARCSECH( SECHAL )
00008      TANHAL  = TANH( AL )
00009      COTHAL  = 1./TANHAL
00010      RNUMJ   = EXP( ORD*( TANHAL-AL ) )
00011      DENJ    = SQRT( 2.*PI*ORD*TANHAL )
00012      BESJ    = ( RNUMJ/DENJ )*( 1.+U1( COTHAL )/ORD+U2( COTHAL )/ORD2 )
00013      ABESJ   = ABS( BESJ )
00014      IF ( ABESJ.LE.1.E-36 ) GO TO 90
00015      RNUMY   = -1./RNUMJ
00016      DENY    = 0.5*DENJ
00017      BESY    = ( RNUMY/DENY )*( 1.-U1( COTHAL )/ORD+U2( COTHAL )/ORD2 )
00018      RBESY   = 1./BESY
00019      GO TO 100
00020  90      BESY    = -1.E+36
00021      RBESY   = 0.00
00022  100     CONTINUE
00023      RETURN
00024      END

00001      SUBROUTINE ABESJD ( ARG,ORD,BESJD )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESJD NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI      = 3.14159265
00005      ORD2     = ORD**2
00006      SECHAL  = ARG/ORD
00007      AL      = ARCSECH( SECHAL )
00008      TANHAL  = TANH( AL )
00009      COTHAL  = 1./TANHAL
00010      RNUMJ   = SQRT( SINH( 2.*AL ) ) * EXP( ORD*( TANHAL-AL ) )
00011      DENJ    = SQRT( 4.*PI*ORD )
00012      BESJD   = ( RNUMJ/DENJ )*( 1.+V1( COTHAL )/ORD+V2( COTHAL )/ORD2 )
00013  100     CONTINUE
00014      RETURN
00015      END

00001      SUBROUTINE ARBESYD ( ARG,ORD,RBESYD )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESJY NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI      = 3.14159265
00005      ORD2     = ORD**2
00006      SECHAL  = ARG/ORD
00007      AL      = ARCSECH( SECHAL )
00008      TANHAL  = TANH( AL )
00009      COTHAL  = 1./TANHAL
00010      RNUMJ   = SQRT( SINH( 2.*AL ) ) * EXP( ORD*( TANHAL-AL ) )
00011      DENJ    = SQRT( 4.*PI*ORD )
00012      BESJD   = ( RNUMJ/DENJ )*( 1.+V1( COTHAL )/ORD+V2( COTHAL )/ORD2 )
00013      ABESJD  = ABS( BESJD )
00014      IF ( ABESJD.LE.1.E-36 ) GO TO 90
00015      RNUMY   = SQRT( SINH( 2.*AL ) ) * EXP( ORD*( AL-TANHAL ) )
00016      DENY    = 0.5*DENJ
00017      BESYD   = ( RNUMY/DENY )*( 1.-V1( COTHAL )/ORD+V2( COTHAL )/ORD2 )
00018      RBESYD  = 1./BESYD

```

```

00019      GO TO 100
00020      90  BESYD  = 1.E+36
00021      RBESYD = 0.00
00022      100  CONTINUE
00023      RETURN
00024      END

00001      SUBROUTINE ABESYR ( A1,A2,ORD,RES )
00002      IF ( A1.GE.ORD ) PRINT *, '1ST ARG.GE.ORD-ABESYR ',
00003      &      'NOT APPLICABLE'
00004      IF ( A1.GE.ORD ) GO TO 100
00005      IF ( A2.GE.ORD ) PRINT *, '2ND ARG.GE.ORD-ABESYR ',
00006      &      'NOT APPLICABLE'
00007      IF ( A2.GE.ORD ) GO TO 100
00008      PI      = 3.14159265
00009      ORD2     = ORD**2
00010      SECHAL1  = A1/ORD
00011      AL1      = ARCSECH(SECHAL1)
00012      TANHAL1  = TANH(AL1)
00013      COTHAL1  = 1./TANHAL1
00014      SECHAL2  = A2/ORD
00015      AL2      = ARCSECH(SECHAL2)
00016      TANHAL2  = TANH(AL2)
00017      COTHAL2  = 1./TANHAL2
00018      SINH2AL2= SINH(2.*AL2)
00019      EXPON    = -ORD*((AL2-TANHAL2)-(AL1-TANHAL1))
00020      DEN      = -SQRT(TANHAL1*SINH2AL2/2.)
00021      SRAT     = (1.-U1(COTHAL1)/ORD+U2(COTHAL1)/ORD2)/
00022      &      (1.-V1(COTHAL2)/ORD+V2(COTHAL2)/ORD2)
00023      RES      = EXP(EXPON)*SRAT/DEN
00024      100  CONTINUE
00025      RETURN
00026      END

00001  C  **** INLET FAN RADIATION SUBROUTINE, E. J. RICE, REVISED 02/21/1998
00002  C  ** CONTAINS PLANE WAVE RADIATION AND IMPROVED TERMINATION LOSS
00003  C  ** THE FOLLOWING "BBRDCFIN" IS THE MAIN SUBROUTINE.  ALL ARGUMENTS
00004  C  ARE INPUTS EXCEPT "NANGLE," "ANGLE," "SPL," "SPLTL" AND WATTS
00005  C  RETURNED TO THE MAIN PROGRAM FOR PRINTING, FILE STORAGE ETC.
00006  C
00007  C  ** THIS VERSION REMOVES THE P**2 OSCILLATIONS BEYOND THE PEAK.  FOUR
00008  C  REGIONS (ETA AND CUT-OFF RATIO) DETERMINE PROPER APPROXIMATION FOR
00009  C  HIGH ANGLE RADIATION.  DEVELOPED 01/30/98.
00010
00011      SUBROUTINE BBRDCFIN(TTOT,PTOT,DISTANCE,
00012      1ISIDELN,DIAM,ALIP,BLIP,FMACHI,FMACHS,NCOF,WATTSCOF,ETAI,
00013      2DELANG,NANGLE,ANGLE,SPL,SPLTL,WATTS,WATTRAN)
00014  C
00015      DIMENSION ANGLE(200),SPL(200),SPLTL(200),WATTSCOF(200),
00016      1COFRAT(200),PSQTOT(200),PSQTLOS(200),PSQRADT(200)
00017
00018      COMMON FMSQ,FM1,BETA,COFBETIN,CFBTINSQ,GDEN,HDEN,
00019      1PSQPK,PSIC,AC,BC,CC,IREG
00020
00021  C
00022  C  ** SUBROUTINES REQUIRED "LIPEF3" AND "PSQGCOF"
00023  C
00024  C  ***** DEFINITION OF SUBROUTINE ARGUMENTS *****
00025  C
00026  C  TTOT      ABSOLUTE TEMPERATURE, (DEGREES RANKINE)

```

```

00027 C PTOT      ABSOLUTE PRESSURE, (PSIA)
00028 C DISTANCE  RADIUS OR SIDELINE DISTANCE OF MICROPHONE ARRAY, (FT.)
      C ISIDELN = 0 FOR CONSTANT RADIUS, = 1 FOR CONSTANT SIDELINE CALCULATION
      C DIAM     INLET DUCT DIAMETER, (INCHES)
      C ALIP     MAJOR OR AXIAL RUNNING DIMENSION OF ELLIPTIC INLET LIP (INCHES)
      C BLIP     MINOR OR TRANSVERSE DIMENSION OF ELLIPTIC INLET LIP (INCHES)
00033 C FMACH     INLET MACH NUMBER, NEGATIVE FOR INLET
00034 C FMACHS    SURROUNDING MACH NUMBER, ALSO NEGATIVE FOR INLET
00035 C NCOF      NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
00036 C WATTSCOF  VECTOR WITH ACOUSTIC POWER FOR EACH BIN, (WATTS)
00037 C ETA       FREQUENCY PARAMETER, (DUCT DIAMETER)/(SOUND WAVELENGTH)
      C DELANG   ANGLE INCREMENT DESIRED ON FAR-FIELD RADIAL TRAVERSE, (DEGREES)
00039 C NANGLE    NUMBER OF ANGLES USED IN CALCULATION, DETERMINED BY DELANG AND
      C          THE MAXIMUM ANGLE OF 90 DEGREES
00041 C ANGLE     VECTOR OF ANGLES ON THE FAR-FIELD TRAVERSE, (DEGREES)
00042 C SPL       THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
00043 C           "IGNORING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
00044 C           2*10**(-5) NEWTONS/METER**2
00045 C SPLTL     THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
      C          "INCLUDING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
      C          2*10**(-5) NEWTONS/METER**2
00048 C WATTS     SUM OF ACOUSTIC POWER IN ALL THE CUT-OFF RATIO BINS, (WATTS)
00049 C WATTRAN   SUM OF TRANSMITTED ACOUSTIC POWER, ALL BINS, (WATTS)
00050
00051          FMACH = FMACHI
00052          ETA = ETAI
00053          DRAD = 0.5*DIAM
00054          PI = 3.1415927
00055          AREAD = PI*DRAD**2
00056          ABELEX = AREAD+2.0*PI*BLIP*(BLIP+0.5*PI*DRAD)
00057          FMBELEX =FMACH*AREAD/ABELEX
00058          ETABELEX = ETA*(DIAM+2.0*BLIP)/DIAM
00059
      C CALCULATE COEF. (PLANE WAVE) INTEGRATION OF RAD. DIRECTIVITY, MACH=0
      IF(ETABELEX.LT.1.0) THEN
00062          ACOEFPW = 0.741697+3.190822*ETABELEX**2.650078
00063          GO TO 14
00064          END IF
00065          ACOEFPW = 3.932518*ETABELEX**1.96285
00066      14 CONTINUE
00067 C
00068
00069          PSQCOEFP = ACOEFPW
00070
00071
00072          QM = 1.0+0.2*FMACHS**2
00073          TSUR = TTOT/QM
00074          PSUR = PTOT/QM**3.5
00075
00076          SONIC = 49.0422*SQRT(TSUR)
00077          RHO = 144.*PSUR/(53.3*TSUR)
00078          POWCON = 8.36424*RHO*SONIC
00079
00080          ETAEXP = ETA**1.08156
00081          WATINFIX = (1.0+1.9036*ETAEXP)/(PI*0.71385*ETAEXP)
00082
00083 C ***** DIMENSIONS, SONIC (FT/SEC), RHO (LBm/FT**3) *****
00084
      C ***** NOTE!! THIS VERSION CALCULATES TO 178 DEGREES FROM INLET AXIS

```

```

00087      NANGLE = 178.0/DELANG
00088      DO 5 I=1,NANGLE
00089      FI = I
00090      ANGLE(I) = FI*DELANG
00091      5 CONTINUE
00092
00093      FMSQ = FMACH**2
00094      FM1 = 1.0-FMSQ
00095      BETA = SQRT(FM1)
00096
00097      ACOEF = 0.7/ETA
00098
00099      FCOF = NCOF
00100      FCOFINV = 1./FCOF
00101      FCOFIND2 = 0.5/FCOF
00102  C ***** SET UP CUT-OFF RATIOS IN THE DUCT *****
00103      COFSQPR = 1.0
00104      DO 20 I=1,NCOF
00105      COFRAT(I) = SQRT(1.0/(COFSQPR-FCOFIND2))
00106      COFSQPR = COFSQPR-FCOFINV
00107      20 CONTINUE
00108
00109  C ***** INITIALIZE P**2 AT EACH FAR-FIELD ANGLE *****
00110  C
00111      DO 10 I=1,NANGLE
00112      PSQRADT(I) = 0.0
00113      PSQTLOS(I) = 0.0
00114      PSQTOT(I) = 0.0
00115      10 CONTINUE
00116  C
00117  C ***** START LOOP ON CUT-OFF RATIO *****
00118  C
00119      WATTS = 0.0
00120      WATTRAN = 0.0
00121
00122      DO 70 J=1,NCOF
00123      WATTS = WATTS+WATTSCOF(J)
00124      POWCOEF = POWCON*WATTSCOF(J)
00125
00126  C *****
00127  C !!!!!!! VERY IMPORTANT! FOLLOWING DEFINES PLANE WAVE REGIME !!!!!
00128  C *****
00129
00130  C ***** DETERMINE IF PLANE WAVE RADIATION ALGORITHM SHOULD BE USED
00131  C ***** IF PLANE WAVE IPW = 1 (YES) FOR COFRATD>=3 OR LAST BIN
00132      IPW = 0
00133      IF(COFRAT(J).GE.3.0.OR.J.EQ.NCOF) IPW=1
00134  C
00135  C ***** PROGRAM SPITS HERE. IT WILL CONTINUE ON FOR NON-PLANE
00136  C ***** WAVES AND WILL JUMP FOR THE PLANE WAVE
00137
00138      IF(IPW.EQ.1) GO TO 45
00139
00140  C ***** NOW IN "NON-PLANE" WAVE, RADIAL OR SPINNING MODE REGIME !!
00141
00142
00143
00144      XID = COFRAT(J)
00145      FRAC = 0.85
00146      IF(XID.LE.2.5) FRAC=1.0-0.1*(XID-1.0)

```

```

00147 C
00148 C ** FOLLOWING ALLOWS SHARP EDGE OR UNFLANGED DUCT APPROXIMATION
00149 BTHK = BLIP/DIAM
00150 IF(BTHK.LT.0.01.OR.XID.GT.10.0) THEN
00151 XIBEL = XID
00152 FMBEL = FMACH
00153 DIAMBEL = DIAM
00154 XDARAD = 1.0
00155 GO TO 23
00156 END IF
00157 C
00158 CALL LIPEF3(XID,XIBEL,FMACH,FMBEL,DIAMBEL,DIAM,ALIP,BLIP,XDARAD,FR
00159 IAC)
00160 IF(FMACH.NE.0.0) THEN
00161 FMBEL = FMACH*ABS(FMBEL/FMACH)
00162 GO TO 23
00163 END IF
00164 FMBEL = 0.0
00165 C
00166 C
00167 23 CONTINUE
00168 ETABEL = ETA*DIAMBEL/DIAM
00169 FMSQ = FMBEL**2
00170 FM1 = 1.0-FMSQ
00171 BETA = SQRT(FM1)
00172
00173 COF = XIBEL
00174 COFINV = 1.0/XIBEL
00175 COFINVSQ = COFINV**2
00176 COFBETIN = 1.0/(XIBEL*BETA)
00177 CFBTINSQ = COFBETIN**2
00178 COFM1 = 1.0-COFINVSQ
00179 COFSQRT = SQRT(COFM1)
00180
00181 A90 = 2.0*(ACOE+COFSQRT)/(ACOE+1.0)
00182 PSQCOEF = A90*(1.0-FMSQ*COFM1)**1.5/BETA
00183 PSQCOEF = PSQCOEF*WATINFI
00184
00185 GDEN = (1.0+COFSQRT)**2
00186 COSPK = BETA*COFSQRT/SQRT(1.0-FMSQ*COFM1)
00187 PSIPK = ACOS(COSPK)*180.0/PI
00188 HDEN = 1.0-FMBEL*COSPK
00189
00190
00191 C ***** CALCULATE TRANSMISSION LOSS IN NON-PLANE WAVE REGION *****
00192
00193 QF = PI*ETABEL*(1.0-1.0/XIBEL)
00194 QF15SQ = (QF-1.5)**2
00195 RADREC = 1.135*EXP(-0.29*(QF+0.18)**2)
00196
00197 IF(QF.LE.1.5) THEN
00198 RADRES = 1.5*EXP(-0.2124*QF15SQ)
00199 GO TO 53
00200 END IF
00201
00202 RADRES = 1.0+0.5*EXP(-0.5338*QF15SQ)
00203 53 CONTINUE
00204
00205 TAU = SQRT(1.0-1.0/XIBEL**2)
00206 TPM = TAU+FMBEL

```

```

00207      TTM = TAU*FMBEL
00208      QDEN = (RADRES*TPM+TTM+1.0)**2+(RADREC*TPM)**2
00209      QNUM = (RADRES+FMBEL)*(RADRES*FMBEL+1.0)+FMBEL*RADREC**2
00210      TLCF = 4.0*TAU*QNUM/QDEN
00211
00212      IF(TLCF.GT.1.0) TLCF=1.0
00213      IF(TLCF.LT.0.0) TLCF=0.0001
00214
00215 C ***** FINISHED WITH TRANSMISSION LOSS AT CURRENT CUT-OFF RATIO **
00216
00217 C ***** CALCULATE TRANSMITTED POWER *****
00218 C
00219      WATTRAN = WATTRAN+TLCF*WATTSCOF(J)
00220 C
00221 C ***** CALCULATE P**2 COEF. TO ACCOUNT FOR WATTS.
00222 C      TO ACCOUNT FOR MIC. RADIUS, DIVIDE BY DISTANCE**2 AT PSQ BELOW.
00223 C
00224
00225      PSQPK = PSQCOEF*ETABEL*XIBEL/(2.0*BETA)
00226 C
00227 C ***** START SORTING INTO REGIMES TO HANDLE LARGE ANGLES *****
00228 C
00229      IREG = 0
00230      ETAC1 = 0.6*BETA/(1.0-COFINV)
00231      IF(ETABEL.GT.ETAC1) THEN
00232      IREG = 1
00233
00234 C ***** REGION 1, SLOPE AT ARG=PI/2 USED TO PROJECT BEYOND PEAK
00235
00236      EPS = 1.0/(BETA*COF)+0.5/ETABEL
00237      EPSQ = EPS**2
00238      QNUM = 1.0+FMSQ*EPSQ
00239      DEPDPSI = QNUM*SQRT(1.0-FM1*EPSQ)
00240      PSQRATC1 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00241      FDEN = CFBTINSQ-EPSQ
00242      DPSQDPSI = (FDEN+4.0*EPSQ)*PSQRATC1*DEPDPSI/(EPS*FDEN)
00243      SINPSIC1 = EPS/SQRT(QNUM)
00244      PSIC = ASIN(SINPSIC1)*180.0/PI
00245      AC = ALOG(PSQRATC1)
00246      BC = 0.8889*DPSQDPSI/PSQRATC1
00247      BC = BC*PI/180.0
00248      CC = -0.1781*BC
00249      GO TO 50
00250      END IF
00251
00252      ETAC2 = 0.6*BETA*COF
00253      IF(ETABEL.GT.ETAC2) THEN
00254      IREG = 2
00255
00256 C ***** REGION 2, PSQratio AT ARG=-PI/2 USED TO FIT EXPONENTIAL FOR
00257 C      USE BEYOND PEAK
00258
00259      EPS = 1.0/(BETA*COF)-0.5/ETABEL
00260      PSQRATC2 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00261      EPSQ = EPS**2
00262      QDEN = 1.0+FMSQ*EPSQ
00263      SINPSIC2 = EPS/SQRT(QDEN)
00264      PSIC2 = ASIN(SINPSIC2)*180.0/PI
00265      AC = ALOG(PSQRATC2)/(PSIPK-PSIC2)**2
00266      GO TO 50

```

```

00267         END IF
00268
00269
00270
C ***** REGION 3, LOW ETA REGION, PSIPK>60 DEG. FIT EXPONENTIAL AT
00272 C    0.5*PSIPK FOR USE BEYOND PEAK. CODE IREG=3. EXPONENTIAL EQUATION
C    USED IN PSQ SUBROUTINE FOR PSI > PSIPK.
00274
00275         IF(PSIPK.GT.60.0) THEN
00276
00277             IREG = 3
00278
00279             ANGF = 0.5*PSIPK
00280             ANGRAD = ANGF*PI/180.0
00281             SINF = SIN(ANGRAD)
00282             EPS = SINF/SQRT(1.0-FMSQ*SINF**2)
00283             ARG = PI*ETABEL*(COFBETIN-EPS)
00284             SINARG = SIN(ARG)
00285             PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00286             PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00287             AC = ALOG(PSQRATC3)/(PSIPK-ANGF)**2
00288             GO TO 50
00289         END IF
00290
C ***** REGION 4, LOW ETA REGION, PSIPK<60 DEG. FIT EXPONENTIAL AT
C    80 deg. FOR USE BEYOND PEAK. CODE IREG=4. EXPONENTIAL EQUATION
00293 C    USED IN PSQ SUBROUTINE FOR PSI>80 deg.
00294
00295         IREG = 4
00296
00297         ANGF = 80.0
00298         ANGRAD = ANGF*PI/180.0
00299         SINF = SIN(ANGRAD)
00300         EPS = SINF/SQRT(1.0-FMSQ*SINF**2)
00301         ARG = PI*ETABEL*(COFBETIN-EPS)
00302         SINARG = SIN(ARG)
00303         PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00304         PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00305         AC = ALOG(PSQRATC3)/(PSIPK-ANGF)**2
00306
00307     50 CONTINUE
00308
00309 C
00310     DO 25 I=1,NANGLE
00311         ANG = ANGLE(I)
00312         IF(ANG.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00313             PSQRADT(I) = 0.0
00314             PSQTOT(I) = 0.0
00315             PSQTLOS(I) = 0.0
00316             GO TO 25
00317         END IF
00318     C
00319     CALL PSQGCOF(ANG,PSQ,FMBEL,ETABEL,XIBEL,PSIPK)
00320
00321     PSQRAD = PSQ
00322     RAD = DISTANCE
00323     IF(ISIDELN.EQ.1) THEN
00324         RAD = DISTANCE/SIN(ANG*PI/180.0)
00325     END IF
00326     PSQ = PSQ/RAD**2

```

```

00327
00328      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00329      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00330      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF
      C ***** NOTE THAT AN INLET TRANSMISSION LOSS (TLCF) HAS BEEN USED ****
      25 CONTINUE
00333
00334      GO TO 70
00335
00336
00337      45 CONTINUE
00338 C ***** IN PLANE-WAVE CALCULATION PROCEDURE !!!!!!!!!!!!!!!!!!!!!!!
00339
00340
00341 C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
00342      PSQPK = 2.0*PSQCOEFP
00343 C *****
00344
00345      GDEN = 4.0
00346
00347      FMSQEX = FMBELEX**2
00348
      C ***** LOW ETA REGION. FIT EXPONENTIAL AT PSICRPL FOR BEYOND PEAK.
      C ***** IF PSICRPL < 90 CAN'T BE OBTAINED, USE PSICRPL = 90.
00351      ANGF = 90.0
00352      ETACRPL = 0.5*SQRT(1.0-FMSQEX)
00353      SINCRPL = 1.0/SQRT(4.0*ETABELEX**2+FMSQEX)
00354      IF(ETABELEX.GT.ETACRPL) ANGF=ASIN(SINCRPL)*180./PI
00355      PSICRPL = ANGF
00356      ANGFRAD = ANGF*PI/180.0
00357      SINF = SIN(ANGFRAD)
00358      ARG = PI*ETABELEX*SINF/SQRT(1.0-FMSQEX*SINF**2)
00359      SINARG = SIN(ARG)
00360      PSQRATPL = (SINARG/ARG)**2
00361      ACPL = ALOG(PSQRATPL)/ANGF**2
00362
00363 C ***** ACPL IS THE AMPLITUDE OF THE EXPONENTIAL AT PSICRPL *****
00364 C ***** END LOW ETA REGION EXPONENTIAL FIT AT PSICRPL. *****
00365
00366 C ** CALCULATE TRANSMISSION LOSS AT BELMOUTH EXIT CUT-OFF RATIO AND
00367 C ** FREQUENCY PARAMETER.
00368
00369      X = 0.5*(PI*ETABELEX)**2
00370      RADRES = 1.0+X*EXP(-0.325226*X)-
00371      1      EXP(-0.101669*ETABELEX**5.7848)
00372      A = 0.023567
00373      Y = 0.5*PI**2*ETABELEX
00374      RADREC=EXP(-3.574331*ETABELEX**1.957292)*8.*ETABELEX/
00375      13.+A*Y**2/(1.+A*Y**3)
00376      QDEN = (1.0+FMBELEX)**2*((RADRES+1.0)**2+RADREC**2)
00377      TLCF=4.*(RADRES*(1.+FMSQEX)+FMBELEX*(RADRES**2+
00378      1      RADREC**2+1.))/QDEN
00379      IF(TLCF.GT.1.0) TLCF=1.0
00380      IF(TLCF.LT.0.0) TLCF=0.0001
00381
      C ** END TRANSMISSION LOSS CALCULATION FOR PLANE WAVE AT BELLMOUTH EXIT
00384 C ***** CALCULATE TRANSMITTED POWER *****
00385 C
00386      WATTRAN = WATTRAN+TLCF*WATTSCOF(J)
00387 C

```

```

00388
00389      DO 40 I=1,NANGLE
00390      FI = I
00391      ANGDEG = ANGLE(I)
00392      IF(ANGDEG.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00393      PSQRADT(I) = 0.0
00394      PSQTOT(I) = 0.0
00395      PSQTLOS(I) = 0.0
00396      GO TO 40
00397      END IF
00398  C
00399      ANGRAD = ANGDEG*PI/180.0
00400
00401      SINANG = SIN(ANGRAD)
00402      COSANG = COS(ANGRAD)
00403
00404      Q1DEN = SQRT(1.0-FMSQEX*SINANG**2)
00405      Q1 = SINANG/Q1DEN
00406      ARG = PI*ETABELEX*Q1
00407      SINSQNUM = (SIN(ARG))**2
00408      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00409      PSQRAT = 1.0
00410
00411      PSQDEN = ARG**2
00412      IF(PSQDEN.LT.1.E-06.AND.ANGDEG.LE.90.0) GO TO 49
00413
00414
00415  C ** LOW FREQUENCY REGION. FOLLOWING ALLOWS REASONABLE
00416  C ** EXTRAPOLATION BEYOND 90 deg, STARTING AT ARG = PI/2 IF ETA>ETAcrit
00417  C ** OR AT 90 deg. OTHERWISE. USES EXPONENTIAL FIT AT PSI = 0
00418  C ** AND PSI = PSIFIT.
00419
00420      IF(ANGDEG.LT.PSICRPL) GO TO 48
00421
00422      QEXP = ACPL*(ANGDEG)**2
00423      IF(QEXP.LT.-20.) QEXP=-20.
00424
00425      PSQRAT = EXP(QEXP)
00426      GO TO 49
00427
00428
00429 48 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00430 49 CONTINUE
00431
00432      PSQ = PSQRAT*PSQPK*GG
00433
00434      PSQRAD = PSQ
00435  C
00436      RAD = DISTANCE
00437      IF(ISIDELN.EQ.1) THEN
00438      RAD = DISTANCE/SIN(ANGDEG*PI/180.0)
00439      END IF
00440
00441      PSQ = PSQ/RAD**2
00442
00443  C
00444      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00445      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00446      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF
00447

```

```

00448 C ***** NOTE THAT A TRANSMISSION LOSS (TLCF) HAS BEEN USED ***
00449
00450 40 CONTINUE
00451 41 CONTINUE
00452
00453
00454 70 CONTINUE
00455
00456     FNANGLE = NANGLE
00457     SUMWATT = 0.0
00458     DO 75 I=1,NANGLE
00459         ANGRAD = ANGLE(I)*PI/180.0
00460         SUMWATT = SUMWATT+PSQRADT(I)*SIN(ANGRAD)
00461
00462         IF(PSQTOT(I).LT.4.E-08) THEN
00463             SPLTL(I) = 20.0
00464             SPL(I) = 20.0
00465             GO TO 75
00466         END IF
00467         SPLTL(I) = 10.0*ALOG10(PSQTLOS(I))+93.9794
00468         SPL(I) = 10.0*ALOG10(PSQTOT(I))+93.9794
00469     75 CONTINUE
00470
00471     WATTINT = SUMWATT*PI/(POWCON*FNANGLE)
00472     SPLDIF = 10.0*ALOG10(WATTS/WATTINT)
00473
00474     DO 80 I=1,NANGLE
00475         SPLTL(I) = SPLTL(I)+SPLDIF
00476         SPL(I) = SPL(I)+SPLDIF
00477     80 CONTINUE
00478
00479
00480     RETURN
00481     END

00001 C
00002 C *****
00003 C ***** END OF MAIN SUBROUTINE "BBRDCFIN" *****
00004 C ***** MODIFIED 02/21/1998, E. J. RICE *****
00005 C *****
00006 C
00007 C *****
00008 C ** SUBROUTINE FOR CALC PSQ FOR EQUAL ENERGY PER MODE AT AN ANGLE
00009 C ** CUT-OFF RATIO APPROXIMATE EQUATIONS USED, BLOCK BUILD-UP AS IN
00010 C AIAA PAPER 96-1774, EMPIRICAL NORMALIZATION REPLACES FACTOR
00011 C Sqrt(1.-1/XI**2). FOUR REGIONS (ETA AND CUT-OFF RATIO) DETERMINE
00012 C PROPER APPROXIMATION FOR HIGH ANGLE RADIATION. DEVELOPED 01/30/98.
00013 C
00014 C SUBROUTINE PSQGCOF(ANG,PSQ,FMACH,ETA,XI,PSIPK)
00015 C
00016 C COMMON FMSQ,FM1,BETA,COFBETIN,CFBTINSQ,GDEN,HDEN,
00017 C 1PSQPK,PSIC,AC,BC,CC,IREG
00018
00019 C PI = 3.1415927
00020
00021 C ANGRAD = ANG*PI/180.0
00022 C SINANG = SIN(ANGRAD)
00023 C COSANG = COS(ANGRAD)
00024

```

```

00025      Q1DEN = SQRT(1.0-FMSQ*SINANG**2)
00026      Q1 = SINANG/Q1DEN
00027      ARG = PI*ETA*(Q1-COFBETIN)
00028      SINSQNUM = (SIN(ARG))**2
00029      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00030      HH = (1.0-FMACH*COSANG)/HDEN
00031      PSQRAT = 4.0*Q1/(BETA*XI*(Q1+COFBETIN)**2)
00032
00033      PSQDEN = ARG**2
00034      ANGCK = PSIPK+1.0
00035
00036      IF(ANG.GT.ANGCK) GO TO 5
00037 C ***** CHECK FOR 0/0, USE LIMITING CASE IF INDETERMINATE *****
00038 C ***** DON'T CHECK IF ANGLE ABOVE PEAK ANGLE *****
00039      IF(PSQDEN.LT.1.E-06) GO TO 39
00040      5 CONTINUE
00041
00042      IF(ANG.LT.PSIPK) GO TO 38
00043
00044      IF(ANG.GE.PSIC.AND.IREG.EQ.1) THEN
00045
00046 C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00047 C ** USES SLOPE AT ARG = PI/2 TO EXTRAPOLATE TO HIGH ANGLES *****
00048      DANG = ANG-PSIC
00049      QEXP = AC+BC*DANG/(1.0+CC*DANG)
00050      IF(QEXP.LT.-20.) QEXP=-20.
00051
00052      PSQRAT = EXP(QEXP)
00053      GO TO 39
00054      END IF
00055
00056      IF(ANG.GE.PSIPK.AND.IREG.EQ.2) THEN
00057
00058 C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00059 C ** USES EXPONENTIAL FIT AT ARG = -PI/2 TO EXTRAPOLATE TO HIGH ANGLES
00060 C ** ALSO HANDLES LOW FREQUENCY REGIMES 3 AND 4 FOR ANGLES BEYOND PEAK.
00061
00062      QEXP = AC*(ANG-PSIPK)**2
00063      IF(QEXP.LT.-20.) QEXP=-20.
00064
00065      PSQRAT = EXP(QEXP)
00066      GO TO 39
00067      END IF
00068
00069      IF(ANG.GE.PSIPK.AND.IREG.EQ.3) THEN
00070
00071 C ** ONE OF THE LOW FREQUENCY REGIONS, REGION 3. FOLLOWING ALLOWS
00072 C ** REASONABLE EXTRAPOLATION BEYOND 90 deg, STARTING AT PEAK.
00073 C ** USES EXPONENTIAL FIT AT PSI = PSIPK AND 0.5*PSIPK. PSIPK>60 deg.
00074
00075      QEXP = AC*(ANG-PSIPK)**2
00076      IF(QEXP.LT.-20.) QEXP=-20.
00077
00078      PSQRAT = EXP(QEXP)
00079      GO TO 39
00080      END IF
00081
00082 C ** ONLY REGION LEFT, REGION 4, WITH PSIPK<60 deg.
00083 C ** ONE OF THE LOW FREQUENCY REGIONS. FOLLOWING ALLOWS REASONABLE
00084 C ** EXTRAPOLATION BEYOND 90 deg, STARTING THIS TIME AT 80 deg.
00085 C ** USES EXPONENTIAL FIT AT PSI = PSIPK AND 80 deg. PSIPK<60 deg.

```

```

00086
00087     IF(ANG.LT.80.0) GO TO 38
00088
00089     QEXP = AC*(ANG-PSIPK)**2
00090     IF(QEXP.LT.-20.) QEXP=-20.
00091
00092     PSQRAT = EXP(QEXP)
00093     GO TO 39
00094
00095     38 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00096     39 CONTINUE
00097
00098     PSQ = PSQRAT*PSQPK*GG*HH
00099 C
00100
00101     RETURN
00102     END

00001 C *****
00002 C ** END OF PSQGCOF.FOR *****
00003 C *****
00004
00005
00006
00007
00008 C
00009 C *****
C ***** ALLOWS CUT OFF MODES TO PROPAGATE IN BELLMOUTH IF THEY CAN
C ***** BE CUT ON OR PROPAGATING BEFORE THE BELLMOUTH EXIT *****
00012 C *****
00013 C ** SUBROUTINE "LIPEF3" CALCULATES THE EFFECT OF THE INLET LIP OR
00014 C BELLMOUTH ON THE INLET FAR-FIELD RADIATION, SECOND MODEL
00015 C *****
00016 C *****
00017 C
00018     SUBROUTINE LIPEF3(XID,XIBEL,FMACK,FMBEL,DBEL,DIAM,ALIP,BLIP,XDARAD
00019     1,FRAC)
00020 C
00021 C ***** DEFINITION OF VARIABLES *****
00022 C
00023 C     ALIP = MINOR AXIS OF THE ELLIPTICAL INLET LIP, INCHES
00024 C     BLIP = MAJOR AXIS OF THE ELLIPTICAL INLET LIP, INCHES
00025 C     FOR CIRCULAR ARC, ALIP = BLIP
00026 C     DIAM = DIAMETER OF INLET DUCT, INCHES
00027 C     FMACH= MACH NUMBER OF UNIFORM FLOW IN THE STRAIGHT DUCT SECTION
00028 C     XID = CUT-OFF RATIO OF MODE IN THE STRAIGHT DUCT SECTION
00029 C     XIBEL = CUT-OFF RATIO OF MODE IN THE ELLIPTIC BELLMOUTH AS THE
C     MODE RELEASES AND RADIATES. USE THIS VALUE FOR RADIATION.
C     XDARAD=VALUE OF X/A WHERE RADIATION RELEASES FROM BELLMOUTH
00032 C
C ** AN ITERATION IS REQUIRED TO DETERMINE XIRAD, START THE ITERATION
C
00035     FMACH = FMACK
00036     IF(FMACK.LT.0.0) FMACH=-FMACK
00037     PI = 3.1415927
00038     N = 100
00039     FN = N
00040     RAD = DIAM/2.0
00041     BDA = BLIP/ALIP
00042     ADUCT = PI*DIAM**2/4.0

```

```

00043      FM1 = SQRT(1.0-FMACH**2)
00044      IF(XID.LE.1.0) GO TO 40
00045  C    ** IF THE MODE DOES NOT PROPAGATE IN THE DUCT, THE BELLMOUTH MAY
00046  C    STILL CUT THE MODE ON AND ALLOW PROPAGATION. SKIP DUCT
00047  C    CALCULATIONS BELOW IF DUCT CUT-OFF RATIO LESS THAN UNITY.
00048      QXI = 1.0-1.0/XID**2
00049      QNT = QXI/(1.0-QXI*FMACH**2)
00050      QNT = FM1*SQRT(QNT)
00051  C    ** ANGDUCT IS THE PROPAGATION ANGLE OF THE MODE IN THE STRAIGHT DUCT
00052      ANGDUCT = ACOS(QNT)
00053      TANANG = TAN(ANGDUCT)
00054      TANANG2 = TANANG**2
00055      XDATSQ = TANANG2/(TANANG2+BDA**2)
00056      XDAT = SQRT(XDATSQ)
00057      QNT = 1.0-XDATSQ
00058      RT = RAD+BLIP*(1.0-SQRT(QNT))
00059      XT = XDAT*ALIP
00060      DELX = (RT-FRAC*RAD)/TANANG
00061      XRAD = XT-DELX
00062      IF(DELX.GE.XT) THEN
00063      XIBEL = XID
00064      ANGDUCT = ANGDUCT*180.0/PI
00065      FMBEL = FMACH
00066      DBEL = DIAM
00067
00068  C
00069  C    ** THE BELLMOUTH DOES NOT EFFECT THE RADIATION AT THIS CUTOFF RATIO
00070      GO TO 501
00071      END IF
00072  40 CONTINUE
00073      XDAPR = 0.0
00074      XDIFPR = XRAD
00075      ICALC = 0
00076      DO 50 I=1,N
00077      FI = I
00078      XDA = SQRT(FI/FN)
00079      IF(XDA.GE.1.0) XDA=0.9999
00080      SQXA = SQRT(1.0-XDA**2)
00081      QNT = BDA*XDA/SQXA
00082  C    ***** ANGVAL IS THE SLOPE OF THE BELLMOUTH WALL AT THIS X/A (XDA)
00083      ANGVAL = ATAN(QNT)
00084  C
00085  C    ** CALCULATE THE INCREASED FLOW AREA DUE TO THE BELLMOUTH AT THIS XDA
00086  C
00087      QRB = BLIP*(1.0-SQXA)
00088      RC = QRB/SIN(ANGVAL)
00089      AEX = 2.0*PI*RC*(RAD*ANGVAL+RC*(1.0-COS(ANGVAL)))
00090  C
00091  C    ** AEX IS THE EXTRA FLOW AREA DUE TO THE BELLMOUTH AT THIS XDA
00092      ATOT = AEX+ADUCT
00093      FMBEL = FMACH*ADUCT/ATOT
00094      RADBEL = RAD+QRB
00095      FM1B = SQRT(1.0-FMBEL**2)
00096      XIBEL= XID*RADBEL*FM1/(RAD*FM1B)
00097      QXI = 1.0-1.0/XIBEL**2
00098      QNT = QXI/(1.0-QXI*FMBEL**2)
00099  C    ***** CHECK IF MODE HAS STARTED PROPAGATING IN BELLMOUTH *****
00100      IF(QXI.LT.0.0) GO TO 50
00101      ICALC = ICALC+1
00102      QNT = FM1B*SQRT(QNT)

```

```

C ***** ANGPROP IS THE ANGLE OF PROPAGATION OF THE MODE AT THIS XDA
      ANGPROP = ACOS(QNT)
00105      TANANG = TAN(ANGPROP)
00106      TANANG2 = TANANG**2
00107      XDATSQ = TANANG2/(TANANG2+BDA**2)
00108      XDAT = SQRT(XDATSQ)
00109      QNT = 1.0-XDATSQ
00110      RT = RAD+BLIP*(1.0-SQRT(QNT))
00111      XT = XDAT*ALIP
00112      DELX = (RT-FRAC*RADBEL)/TANANG
00113      XRAD = XT-DELX
00114      XDIF = XRAD-XDA*ALIP
00115      IF(XDIF.LE.0.0) GO TO 55
00116      XDIFPR = XDIF
00117      XDAPR = XDA
00118      50 CONTINUE
00119      IF(ICALC.EQ.0) THEN
00120      WRITE(12,100)
00121      WRITE(*,100)
00122      100 FORMAT(' ***** ALERT ***** ALERT ***** ALERT *****')
00123      WRITE(12,101)
00124      WRITE(*,101)
      101 FORMAT(' ** THIS MODE CAN NOT ESCAPE THE BELLMOUTH EXIT. PLEASE M
      LAKE MODIFICATIONS ** ')
00127      WRITE(12,100)
00128      WRITE(*,100)
00129      XIBEL = .1
00130      GO TO 500
00131      END IF
00132      IF(ICALC.EQ.1) THEN
00133      C ***** MODE CUT-ON ACHIEVED AT BELLMOUTH EXIT *****
00134      XIBEL = 1.0001
00135      GO TO 500
00136      END IF
00137      55 CONTINUE
00138      IF(ICALC.EQ.1) THEN
00139      XDARAD = XDA
00140      GO TO 60
00141      END IF
00142      X2 = XDA
00143      X1 = XDAPR
00144      Y2 = XDIF
00145      Y1 = XDIFPR
00146      DY21 = ABS(Y2-Y1)
00147      IF(DY21.EQ.0.0) THEN
00148      XDARAD = XDAPR
00149      GO TO 60
00150      END IF
00151      XDARAD = (X1*Y2-X2*Y1)/(Y2-Y1)
00152      C
00153      C ** ITERATION DONE, CALCULATE OUTPUTS AT X/A = XDARAD
00154      C
00155      60 CONTINUE
00156      SQXA = SQRT(1.0-XDARAD**2)
00157      QNT = BDA*XDARAD/SQXA
      C ***** ANGVAL IS THE SLOPE OF THE BELLMOUTH WALL AT MODE RADIATION
      ANGVAL = ATAN(QNT)
00160      C
00161      C ** CALCULATE THE FINAL FLOW AREA INCREASE DUE TO THE BELLMOUTH
00162      C

```

```

00163      QRB = BLIP*(1.0-SQXA)
00164      RC = QRB/SIN(ANGWAL)
00165      AEX = 2.0*PI*RC*(RAD*ANGWAL+RC*(1.0-COS(ANGWAL)))
00166      C
00167      C  ** AEX IS THE EXTRA FLOW AREA DUE TO THE BELLMOUTH AT THIS XDA
00168      ATOT = AEX+ADUCT
00169      FMBEL = FMACH*ADUCT/ATOT
00170      RADBEL = RAD+QRB
00171      FM1B = SQRT(1.0-FMBEL**2)
00172      XIBEL= XID*RADBEL*FM1/(RAD*FM1B)
00173      QXI = 1.0-1.0/XIBEL**2
00174      QNT = QXI/(1.0-QXI*FMBEL**2)
00175      QNT = FM1B*SQRT(QNT)
00176      IF(XID.GE.1.0) THEN
00177      QXI = 1.0-1.0/XID**2
00178      QNT = QXI/(1.0-QXI*FMACH**2)
00179      QNT = FM1*SQRT(QNT)
00180      C  ** "ANGDUCT" IS THE PROPAGATION ANGLE OF THE MODE FROM THE DUCT IF
00181      C  BELLMOUTH IS "NOT" CONSIDERED *****
00182      ANGDUCT = ACOS(QNT)*180.0/PI
00183      ELSE
00184      ANGDUCT = 90.0
00185      END IF
00186      500 CONTINUE
00187      DBEL = 2.0*RADBEL
00188      501 CONTINUE
00189      RETURN
00190      END

00001      C
00002      C  *****
00003      C  ***** END OF SUBROUTINE "LIPEF3" *****
00004      C  *****
00005
00006      C
00007      C  **** AFT FAN RADIATION SUBROUTINE, E. J. RICE, REVISED 02/19/1998
00008      C  **** NORMALIZED PLANE WAVE RADIATION INCLUDED HERE.
00009      C  ** THE FOLLOWING "BBRDCFEX" IS THE MAIN SUBROUTINE. ALL ARGUMENTS
00010      C  ARE INPUTS EXCEPT "DJET," "FMACH1," "NANGLE," "ANGLE," "SPL,"
00011      C  "SPLTL," "WATTS," "FMACHN," AND "COFMIN" AND WATTS
00012      C  RETURNED TO THE MAIN PROGRAM FOR PRINTING, FILE STORAGE ETC.
00013      C
00014      C  ** THIS VERSION REMOVES THE P**2 OSCILLATIONS BEYOND THE PEAK. FOUR
00015      C  REGIONS (ETA AND CUT-OFF RATIO) DETERMINE PROPER APPROXIMATION FOR
00016      C  HIGH ANGLE RADIATION. DEVELOPED 01/30/98.
00017
00018      SUBROUTINE BBRDCFEX(TTOT,PTOT,TSUR,PSUR,HTRAT,
00019      1ANOZRAT,DISTANCE,ISIDELN,DDUCT,DJET,FMACHD,FMACH1,FMACH2,
00020      2NCOF,WATTSCOF,DELANG,ETAD,
00021      3NANGLE,ANGLE,SPL,SPLTL,WATTS,WATTRAN,FMACHN,COFMIN)
00022
00023      C
00024      DIMENSION ANGLE(200),SPL(200),SPLTL(200),WATTSCOF(200),
00025      1COFRAT(200),COFRATD(200),COFRATN(200),PSQTOT(200),
00026      2PSQTLOS(200),PSQRADT(200)
00027
00028      C
00029      C  ** SUBROUTINE REQUIRED "CONOZ"
00030      C

```

```

00031 C ***** DEFINITION OF SUBROUTINE ARGUMENTS *****
00032 C
00033 C TTOT      TOTAL TEMPERATURE IN AFT FAN DUCT, (DEGREES RANKINE)
00034 C PTOT      TOTAL PRESSURE IN AFT FAN DUCT, (PSIA)
00035 C TSUR      TOTAL TEMPERATURE IN SURROUNDING MEDIUM, (DEGREES RANKINE)
00036 C PSUR      TOTAL PRESURE IN SURROUNDING MEDIUM, (PSIA)
00037 C HTRAT     AFT FAN DUCT HUB-TIP RATIO
00038 C ANOZRAT   (NOZZLE THROAT AREA)/(FAN DUCT AREA)
00039 C DISTANCE  RADIUS OF MICROPHONE ARRAY OR TRAVERSE, (FT.)
      C ISIDELN = 0 FOR CONSTANT RADIUS, = 1 FOR CONSTANT SIDELINE CALCULATION
      C DDUCT   AFT FAN DUCT OUTSIDE DIAMETER, (INCHES)
00042 C DJET     FINAL JET DIAMETER, (INCHES)
00043 C FMACHD    AFT FAN DUCT MACH NUMBER, POSITIVE FOR EXHAUST
00044 C FMACH1    FINAL JET MACH NUMBER
00045 C FMACH2    MACH NUMBER OF SURROUNDING MEDIUM
00046 C FMACHN    NOZZLE EXIT (THROAT) MACH NUMBER
00047 C NCOF      NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
00048 C WATTSCOF   VECTOR WITH ACOUSTIC POWER FOR EACH BIN, (WATTS)
00049 C FREQ      FREQUENCY OF SOUND, (HERTZ)
      C DELANG  ANGLE INCREMENT DESIRED ON FAR-FIELD RADIAL TRAVERSE, (DEGREES)
00051 C NANGLE    NUMBER OF ANGLES USED IN CALCULATION, DETERMINED BY DELANG AND
      C          THE MAXIMUM ANGLE OF 180 DEGREES
00053 C ANGLE     VECTOR OF ANGLES ON THE FAR-FIELD TRAVERSE, CONVERTED TO
00054 C          THAT MEASURED FROM THE ENGINE "INLET" AXIS, (DEGREES)
00055 C SPL       THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
00056 C          "IGNORING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
00057 C           $2 \times 10^{(-5)}$  NEWTONS/METER**2
00058 C SPLTL    THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
      C          "INCLUDING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
      C           $2 \times 10^{(-5)}$  NEWTONS/METER**2
00061 C WATTS     SUM OF ACOUSTIC POWER IN ALL THE CUT-OFF RATIO BINS, (WATTS)
00062 C WATTRAN    SUM OF TRANSMITTED ACOUSTIC POWER IN ALL BINS, (WATTS)
00063 C COFMIN    THE MINIMUM CUT-OFF RATIO BELOW WHICH COMPLETE REFLECTION
00064 C          OCCURS DUE TO REFRACTION THROUGH THE SLIP LAYER
00065 C
00066          PI = 3.1415927
00067          QAFP = 1.0+0.328766*ETAD**1.702882
00068          AFPOWFAC = 1.741*(QAFP+1.274989*ETAD**2)/(PI*ETAD*QAFP)
00069
      C CALCULATE COEF. (PLANE WAVE) INTEGRATION OF RAD. DIRECTIVITY, MACH=0
      IF(ETAD.LT.1.0) THEN
00072          ACOEFPW = 1.733303+5.30259*ETAD**2.28937
00073          GO TO 14
00074          END IF
00075          ACOEFPW = 7.035893*ETAD**1.773669
00076          14 CONTINUE
00077 C
00078 C CALC. COEF. FOR INTEGRATION WITH AFT SHEAR LAYERS, PLANE WAVE
00079 C
00080          AFPOWFPW = (1.0+0.127683*ETAD)/(3.0+0.137590*ETAD)
00081          PSQCOEFP = ACOEFPW*AFPOWFPW*(1.0+2.6557*ETAD)/(1.0515+3.8508*ETAD)
00082          PSQCOEFP = 1.3704*PSQCOEFP
00083 C *****
00084
00085          FCOF = NCOF
00086          FCOFINV = 1./FCOF
00087          FCOFIND2 = 0.5/FCOF
00088 C ***** SET UP CUT-OFF RATIOS IN THE DUCT *****
00089          COFSQPR = 1.0
00090          DO 20 I=1,NCOF

```

```

00091      COFRATD(I) = SQRT(1.0/(COFSQPR-FCOFIND2))
00092      COFSQPR = COFSQPR-FCOFINV
00093      20 CONTINUE
00094
00095      TDUCT = TTOT/(1.0+0.2*FMACHD**2)
00096      CDUCT = 49.0421*SQRT(TDUCT)
00097
00098      QSUR = 1.0+0.2*FMACH2**2
00099      TSTS = TSUR/QSUR
00100      PSTS = PSUR/QSUR**3.5
00101      CSUR = 49.0421*SQRT(TSTS)
00102      RHOSUR = 144.0*PSTS/(53.3*TSTS)
00103
00104      C
00105      C ***** DETERMINE NOZZLE FLOW PROPERTIES *****
00106
00107      CALL CONOZ(TTOT,PTOT,PSTS,HTRAT,ANOZRAT,DDUCT,FMACHD,FMACH1,CJET,D
00108      1JET,TJET,PNOZ,DNOZ,CNOZ,FMACHN)
00109
00110      CRAT = CJET/CSUR
00111
00112      ETA = ETAD*DJET*CDUCT/(DDUCT*CJET)
00113      ETAN= ETAD*DNOZ*CDUCT/(DDUCT*CNOZ)
00114
00115      RATCFNOZ = DNOZ*CDUCT*SQRT(1.0-FMACHD**2)/(DDUCT*CNOZ*
00116      1SQRT(1.0-FMACHN**2))
00117
00118      RATCFJET = DJET*CDUCT*SQRT(1.0-FMACHD**2)/(DDUCT*CJET*
00119      1SQRT(1.0-FMACH1**2))
00120
00121
00122      FMSQ1 = FMACH1**2
00123      FM11 = 1.0-FMSQ1
00124      BETA1 = SQRT(FM11)
00125      FMSQ2 = FMACH2**2
00126      C
00127      C ***** ESTABLISH CUT-OFF LIMITS FOR COMPLETE REFLECTION
00128      COFMIN = 1.0
00129      IF(FMACH1.EQ.0.0) GO TO 15
00130
00131      CKM2 = 1.0-CRAT*FM11/FMACH1
00132      IF(FMACH2.LT.CKM2) THEN
00133      COSPHIL = -CRAT/(1.0+CRAT*FMACH1-FMACH2)
00134      PHIL = ACOS(COSPHIL)
00135      SINPHIL = SIN(PHIL)
00136      DEN = SQRT(1.0+FMSQ1+2.0*FMACH1*COSPHIL)
00137      COSPSIL = (COSPHIL+FMACH1)/DEN
00138      SINPSIL = SINPHIL/DEN
00139      COFMIN = SQRT(FM11+FMSQ1*COSPSIL**2)/(BETA1*SINPSIL)
00140      C      COFMIN = (1.0+FMACH1*COSPHIL)/(BETA1*SINPHIL)
00141      END IF
00142      15 CONTINUE
00143      C *****
00144
00145      NANGLE = 180.0/DELANG-1
00146      DO 5 I=1,NANGLE
00147      FI = I
00148      ANGLE(I) = FI*DELANG
00149      5 CONTINUE
00150

```

```

00151 C ***** PART OF EMPIRICAL COEFFICIENT TAKING PLACE OF RADICAL
00152 C ***** SQRT(1-1/COF**2) IN P**2 COEFFICIENT WHICH WOULD NOT BE
00153 C ***** ANY GOOD AT CUT-OFF. FOR NON-PLANE WAVES.
00154
00155         ACOEF = 0.7/ETA
00156
00157 C ** RECALL, THE CUT-OFF RATIOS IN THE DUCT ARE INPUT HERE AS "COFRATD"
00158
00159 C ***** CALCULATE CUT-OFF RATIOS IN THE NOZZLE AND THE JET *****
00160         DO 22 I=1,NCOF
00161             COFRATN(I) = RATCFNOZ*COFRATD(I)
00162             COFRAT(I) = RATCFJET*COFRATD(I)
00163         22 CONTINUE
00164 C
00165 C ***** INITIALIZE P**2 AT EACH FAR-FIELD ANGLE *****
00166 C
00167         DO 10 I=1,NANGLE
00168             PSQRADT(I) = 0.0
00169             PSQTLOS(I) = 0.0
00170             PSQTOT(I) = 0.0
00171         10 CONTINUE
00172 C
00173 C ***** START LOOP ON CUT-OFF RATIO *****
00174 C
00175         POWCON = 8.36424*RHOSUR*CSUR
00176         WATTS = 0.0
00177         WATTRAN = 0.0
00178         DO 70 J=1,NCOF
00179             WATTS = WATTS+WATTSCOF(J)
00180 C
00181 C ** DETERMINE IF COFRAT IS WITHIN COMPLETE REFLECTION RANGE DUE TO
00182 C ** REFRACTION THROUGH THE SLIP LAYER OR IF COMPLETE REFLECTION
00183 C ** OCCURS AT THE NOZZLE THROAT
00184 C
00185         IF(COFRAT(J).LT.COFRATN(J).LE.1.0) GO TO 70
00186
00187
00188 C *****
00189 C !!!!!!! VERY IMPORTANT! FOLLOWING DEFINES PLANE WAVE REGIME !!!!!
00190 C *****
00191
00192 C ***** DETERMINE IF PLANE WAVE RADIATION ALGORITHM SHOULD BE USED
00193 C ***** IF PLANE WAVE IPW = 1 (YES) FOR COFRATD>=3 OR LAST BIN
00194         IPW = 0
00195         IF(COFRATD(J).GE.3.0.OR.J.EQ.NCOF) IPW=1
00196
00197
00198
00199
00200 C ** CALCULATE TRANSMISSION LOSS AT NOZZLE THROAT CUT-OFF RATIO AND
00201 C ** FREQUENCY PARAMETER.
00202
00203         FMSQN = FMACHN**2
00204
00205         IF(IPW.EQ.1) THEN
00206
00207 C ***** IN PLANE-WAVE REGIME *****
00208
00209             X = 0.5*(PI*ETAN)**2
00210             RADRES = 1.0+X*EXP(-0.325226*X)-EXP(-0.101669*ETAN**5.7848)
00211             A = 0.023567

```

```

00212      Y = 0.5*PI**2*ETAN
00213      RADREC=EXP(-3.574331*ETAN**1.957292)*8.*
00214      1      ETAN/3.+A*Y**2/(1.+A*Y**3)
00215      QDEN = (1.0+FMACHN)**2*((RADRES+1.0)**2+RADREC**2)
00216      TLCF = 4.*(RADRES*(1.+FMSQN)+FMACHN*(RADRES**2+
00217      1      RADREC**2+1.))/QDEN
00218      GO TO 55
00219      END IF
00220
00221  C ***** IN NON-PLANE WAVE, RADIAL OR SPINNING MODE REGION *****
00222
00223      QF = PI*ETAN*(1.0-1.0/COFRATN(J))
00224      QF15SQ = (QF-1.5)**2
00225      RADREC = 1.135*EXP(-0.29*(QF+0.18)**2)
00226
00227      IF(QF.LE.1.5) THEN
00228      RADRES = 1.5*EXP(-0.2124*QF15SQ)
00229      GO TO 53
00230      END IF
00231
00232      RADRES = 1.0+0.5*EXP(-0.5338*QF15SQ)
00233  53 CONTINUE
00234
00235      TAU = SQRT(1.0-1.0/COFRATN(J)**2)
00236      TPM = TAU+FMACHN
00237      TTM = TAU*FMACHN
00238      QDEN = (RADRES*TPM+TTM+1.0)**2+(RADREC*TPM)**2
00239      QNUM = (RADRES+FMACHN)*(RADRES*FMACHN+1.0)+FMACHN*RADREC**2
00240      TLCF = 4.0*TAU*QNUM/QDEN
00241
00242  C ***** CALCULATE TRANSMITTED POWER *****
00243  C
00244  55 CONTINUE
00245      WATTRAN = WATTRAN+TLCF*WATTSCOF(J)
00246  C
00247  C ***** CALCULATE P**2 COEF. TO ACCOUNT FOR WATTS *****
00248  C      TO ACCOUNT FOR MIC. RADIUS, DIVIDE BY DISTANCE**2 AT PSQ BELOW.
00249  C
00250      POWCOEF = POWCON*WATTSCOF(J)
00251  C
00252  C ***** PROGRAM SPITS HERE. IT WILL CONTINUE ON FOR NON-PLANE
00253  C ***** WAVES AND WILL JUMP FOR THE PLANE WAVE
00254
00255      IF(IPW.EQ.1) GO TO 45
00256
00257  C ***** NOW IN "NON-PLANE" WAVE, RADIAL OR SPINNING MODE REGIME !!
00258
00259      COFBETIN = 1.0/(COFRAT(J)*BETA1)
00260      COFINV = 1.0/COFRAT(J)
00261      COFINVSQ = COFINV**2
00262      COFM1 = 1.0-COFINVSQ
00263      EP = SQRT(COFM1)
00264      GDEN = (1.0+EP)**2
00265
00266  C ***** HERE IS REMAINDER OF EMPIRICAL COEFFICIENT USING "ACOE" ABOVE
00267
00268      A90 = 2.0*(ACOE+EP)/(ACOE+1.0)
00269
00270  C ***** THEORETICAL NORMALIZATION COEFFICIENT WITH FLOW ATTACHED TO A90
00271  C ** INCLUDES INTEGRATED POWER NORMALIZATION "AFPOWFAC" (EMPIRICAL) *
00272

```

```

00273      PSQCOEF = AFPOWFAC*A90*(1.0-FMSQ1*COFM1)**1.5/BETA1
00274
00275      COSPK1 = BETA1*EP/SQRT(1.0-FMSQ1*COFM1)
00276      ANGPK1 = ACOS(COSPK1)
00277      C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 1, PSIPK1 (DEGREES)
00278      PSIPK1 = ANGPK1*180.0/PI
00279      SINPK1 = SIN(ANGPK1)
00280      C ***** PHASE VELOCITY VECTOR ANGLE FOR PEAK IN REGION 1 (JET)
00281      SIN2 = SINPK1**2
00282      COSPHI1 = -FMACH1*SIN2+COSPK1*SQRT(1.0-FMSQ1*SIN2)
00283      C
00284      C ***** PHI1 TO PHI2 ACROSS SLIP LAYER
00285      C ***** PHASE VELOCITY ANGLE IN REGION 2 (SURROUNDINGS)
00286
00287      COSPHI2 = COSPHI1/(CRAT+(CRAT*FMACH1-FMACH2)*COSPHI1)
00288
00289      C ***** PHASE TO GROUP VELOCITY ANGLES IN REGION 2 (SURROUNDINGS)
00290
00291      COSPSI2 = (COSPHI2+FMACH2)/SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00292      PSI2RAD = ACOS(COSPSI2)
00293      SINPSI2 = SIN(PSI2RAD)
00294
00295      C ***** ANGLE CHANGE ACOUSTIC POWER CORRECTION *****
00296
00297      FREFRCT = SINPK1/SINPSI2
00298
00299      C *****
00300
00301      Q22NUM = SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00302
00303      COSPSPK2 = (COSPHI2+FMACH2)/Q22NUM
00304      ANGPK2 = ACOS(COSPSPK2)
00305
00306      C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 2, PSIPK2 (DEGREES)
00307      PSIPK2 = ANGPK2*180.0/PI
00308
00309      C GROUP VELOCITY VECTOR ANGLE CHANGE, REG. 1 TO REG. 2, DELPSI (DEG.)
00310      DELPSI = PSIPK2-PSIPK1
00311      C *****
00312
00313      SIN2PK2 = SIN(PSIPK2*PI/180.0)
00314
00315      C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
00316      PSQPK = PSQCOEF*FREFRCT*ETA*COFRAT(J)/(2.0*BETA1)
00317      C *****
00318
00319
00320      C ***** START SORTING INTO REGIMES TO HANDLE ANGLES BEYOND PEAK *****
00321      C
00322      COF = COFRAT(J)
00323      IREG = 0
00324      ETAC1 = 0.6*BETA1/(1.0-COFINV)
00325      IF(ETA.GT.ETAC1) THEN
00326      IREG = 1
00327
00328      C ***** REGION 1, SLOPE AT ARG=PI/2 USED TO PROJECT BEYOND PEAK
00329
00330      EPS = 1.0/(BETA1*COF)+0.5/ETA
00331      EPSQ = EPS**2
00332      QNUM = 1.0+FMSQ1*EPSQ
00333      DEPDPSI = QNUM*SQRT(1.0-FM11*EPSQ)

```

```

00334      PSQRATC1 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00335      FDEN = CFBTINSQ-EPSQ
00336      DPSQDPSI = (FDEN+4.0*EPSQ)*PSQRATC1*DEPDPSI/(EPS*FDEN)
00337      SINPSIC1 = EPS/SQRT(QNUM)
00338      PSIC = ASIN(SINPSIC1)*180.0/PI
00339      AC = ALOG(PSQRATC1)
00340      BC = 0.8889*DPSQDPSI/PSQRATC1
00341      BC = BC*PI/180.0
00342      CC = -0.1781*BC
00343      GO TO 50
00344      END IF
00345
00346      ETAC2 = 0.6*BETA1*COF
00347      IF(ETA.GT.ETAC2) THEN
00348          IREG = 2
00349
00350      C ***** REGION 2, PSQratio AT ARG=-PI/2 USED TO FIT EXPONENTIAL FOR
00351      C          USE BEYOND PEAK
00352
00353      EPS = 1.0/(BETA1*COF)-0.5/ETA
00354      PSQRATC2 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00355      EPSQ = EPS**2
00356      QDEN = 1.0+FMSQ1*EPSQ
00357      SINPSIC2 = EPS/SQRT(QDEN)
00358      PSIC2 = ASIN(SINPSIC2)*180.0/PI
00359      AC = ALOG(PSQRATC2)/(PSIPK1-PSIC2)**2
00360      GO TO 50
00361      END IF
00362
00363
00364      C ***** REGION 3, LOW ETA REGION, PSIPK1>60 DEG. FIT EXPONENTIAL AT
00365      C          0.5*PSIPK1 FOR USE BEYOND PEAK. CODE IREG=3. EXPONENTIAL EQUATION
00366      C          USED IN PSQ SUBROUTINE FOR PSI > PSIPK1.
00367
00368
00369      IF(PSIPK1.GT.60.0) THEN
00370
00371          IREG = 3
00372
00373          ANGF = 0.5*PSIPK1
00374          ANGRAD = ANGF*PI/180.0
00375          SINF = SIN(ANGRAD)
00376          EPS = SINF/SQRT(1.0-FMSQ1*SINF**2)
00377          ARG = PI*ETA*(COFBETIN-EPS)
00378          SINARG = SIN(ARG)
00379          PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00380          PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00381          AC = ALOG(PSQRATC3)/(PSIPK1-ANGF)**2
00382          GO TO 50
00383          END IF
00384
00385      C ***** REGION 4, LOW ETA REGION, PSIPK1<60 DEG. FIT EXPONENTIAL AT
00386      C          80 deg. FOR USE BEYOND PEAK. CODE IREG=4. EXPONENTIAL EQUATION
00387      C          USED IN PSQ SUBROUTINE FOR PSI>80 deg.
00388
00389          IREG = 4
00390
00391          ANGF = 80.0
00392          ANGRAD = ANGF*PI/180.0
00393          SINF = SIN(ANGRAD)

```

```

00394      EPS = SIN(PI/2)/SQRT(1.0-FMSQ1*SIN**2)
00395      ARG = PI*ETA*(COFBETIN-EPS)
00396      SINARG = SIN(ARG)
00397      PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00398      PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00399      AC = ALOG(PSQRATC3)/(PSIPK1-ANGF)**2
00400
00401      50 CONTINUE
00402      C
      C ***** FINISHED WITH 4 REGION DEFINITIONS, START PSQ CALCULATION
00405      DO 25 I=1,NANGLE
00406      FI = I
00407      ANGDEG2 = ANGLE(I)
00408
00409      IF(ANGDEG2.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00410      PSQRADT(I) = 0.0
00411      PSQTOT(I) = 0.0
00412      PSQTLOS(I) = 0.0
00413      GO TO 25
00414      END IF
00415      C
00416      ANGDEG1 = ANGDEG2-DELPsi
00417      ANG = ANGDEG1
00418      ANGRAD1 = ANGDEG1*PI/180.0
00419      IF(ANGDEG1.LT.0.0) GO TO 25
00420      SINANG = SIN(ANGRAD1)
00421      COSANG = COS(ANGRAD1)
00422
00423      Q1DEN = SQRT(1.0-FMSQ1*SINANG**2)
00424      Q1 = SINANG/Q1DEN
00425      ARG = PI*ETA*(Q1-COFBETIN)
00426      SINSQNUM = (SIN(ARG))**2
00427      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00428      PSQRAT = 4.0*Q1/(BETA1*COFRAT(J)*(Q1+COFBETIN)**2)
00429
00430      PSQDEN = ARG**2
00431      ANGCK = PSIPK1+1.0
00432      IF(ANG.GT.ANGCK) GO TO 6
00433      C ***** CHECK FOR 0/0, USE LIMITING CASE IF INDETERMINATE *****
00434      C ***** DON'T CHECK IF ANGLE ABOVE PEAK ANGLE *****
00435      IF(PSQDEN.LT.1.E-06) GO TO 39
00436      6 CONTINUE
00437
00438      IF(ANG.LT.PSIPK1) GO TO 38
00439      IF(ANG.GE.PSIC.AND.IREG.EQ.1) THEN
00440
00441      C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00442      C ** USES SLOPE AT ARG = PI/2 TO EXTRAPOLATE TO HIGH ANGLES *****
00443      DANG = ANG-PSIC
00444      QEXP = AC+BC*DANG/(1.0+CC*DANG)
00445      IF(QEXP.LT.-20.) QEXP=-20.
00446
00447      PSQRAT = EXP(QEXP)
00448      GO TO 39
00449      END IF
00450
00451      IF(ANG.GE.PSIPK1.AND.IREG.EQ.2) THEN
00452
00453      C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
      C ** USES EXPONENTIAL FIT AT ARG = -PI/2 TO EXTRAPOLATE TO HIGH ANGLES

```

```

00456      QEXP = AC*(ANG-PSIPK1)**2
00457      IF(QEXP.LT.-20.) QEXP=-20.
00458
00459      PSQRAT = EXP(QEXP)
00460      GO TO 39
00461      END IF
00462
00463      IF(ANG.GE.PSIPK1.AND.IREG.EQ.3) THEN
00464
00465      C  ** ONE OF THE LOW FREQUENCY REGIONS, REGION 3. FOLLOWING ALLOWS
00466      C  ** REASONABLE EXTRAPOLATION BEYOND 90 deg, STARTING AT PEAK.
00467      C  ** USES EXPONENTIAL FIT AT PSI = PSIPK1 AND 0.5*PSIPK1. PSIPK1>60 deg

00469      QEXP = AC*(ANG-PSIPK1)**2
00470      IF(QEXP.LT.-20.) QEXP=-20.
00471
00472      PSQRAT = EXP(QEXP)
00473      GO TO 39
00474      END IF
00475
00476      C  ** ONLY REGION LEFT, REGION 4, WITH PSIPK1<60 deg.
00477      C  ** ONE OF THE LOW FREQUENCY REGIONS. FOLLOWING ALLOWS REASONABLE
00478      C  ** EXTRAPOLATION BEYOND 90 deg, STARTING THIS TIME AT 80 deg.
00479      C  ** USES EXPONENTIAL FIT AT PSI = PSIPK1 AND 80 deg. PSIPK1<60 deg.
00480
00481      IF(ANG.LT.80.0) GO TO 38
00482
00483      QEXP = AC*(ANG-PSIPK1)**2
00484      IF(QEXP.LT.-20.) QEXP=-20.
00485
00486      PSQRAT = EXP(QEXP)
00487      GO TO 39
00488
00489      38 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00490      39 CONTINUE
00491
00492      PSQ = PSQRAT*PSQPK*GG
00493      PSQRAD = PSQ
00494      C
00495      RAD = DISTANCE
00496      IF(ISIDELN.EQ.1) THEN
00497      RAD = DISTANCE/SIN(ANGDEG2*PI/180.0)
00498      END IF
00499
00500      PSQ = PSQ/RAD**2
00501
00502      C
00503      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00504      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00505      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF
00506
00507      C  ***** NOTE THAT A TRANSMISSION LOSS (TLCF) HAS BEEN USED *****
00508
00509      25 CONTINUE
00510      26 CONTINUE
00511      GO TO 70
00512
00513
00514      45 CONTINUE
00515      C  ***** IN PLANE-WAVE CALCULATION PROCEDURE !!!!!!!!!!!!!!!!!!!!!!!
00516

```

```

00517      GDEN = 4.0
00518
00519      IF(FMACH1.EQ.FMACH2) THEN
00520          DELPSI = 0.0
00521          GO TO 27
00522      END IF
00523
00524      COSPK1 = 1.0
00525      ANGPK1 = 0.0
00526      C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 1, PSIPK1 (DEGREES)
00527          PSIPK1 = 0.0
00528          SINPK1 = 0.0
00529      C ***** PHASE VELOCITY VECTOR ANGLE FOR PEAK IN REGION 1 (JET)
00530          SIN2 = SINPK1**2
00531          COSPHI1 = 1.0
00532          PHI1RAD = 0.0
00533          PHI1DEG = 0.0
00534          SINPHI1 = 0.0
00535      C
00536      C ***** PHASE VELOCITY ANGLE IN REGION 2 (SURROUNDINGS)
00537
00538          COSPHI2 = 1.0/(CRAT+CRAT*FMACH1-FMACH2)
00539          PHI2RAD = ACOS(COSPHI2)
00540          PHI2DEG = PHI2RAD*180.0/PI
00541          SINPHI2 = SIN(PHI2RAD)
00542          COSPSI2 = (COSPHI2+FMACH2)/SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00543          PSI2RAD = ACOS(COSPSI2)
00544          PSI2DEG = PSI2RAD*180.0/PI
00545          SINPSI2 = SIN(PSI2RAD)
00546
00547          Q22DEN = SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00548
00549          COSPSPK2 = (COSPHI2+FMACH2)/Q22DEN
00550          ANGPK2 = ACOS(COSPSPK2)
00551          PSIPK2 = ANGPK2*180.0/PI
00552
00553      C GROUP VELOCITY VECTOR ANGLE CHANGE, REG. 1 TO REG. 2, DELPSI (DEG.)
00554          DELPSI = PSIPK2-PSIPK1
00555      C *****
00556
00557      C
00558          27 CONTINUE
00559
00560      C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
00561          PSQPK = 2.0*PSQCOEFP
00562      C *****
00563
00564      C **** PSI10 BELOW IS SMALLEST ANGLE WHERE PLANE WAVE RADIATION, P**2=0
00565      C **** IF PSI10 > 90, PSI10 = 90 IS USED
00566          SINPSI10 = 1.0/SQRT(ETA**2+FMACH1**2)
00567          IF(SINPSI10.LT.1.0) THEN
00568              ANG10 = ASIN(SINPSI10)
00569              PSI10 = ANG10*180.0/PI
00570              COSPSI10 = COS(ANG10)
00571              GO TO 28
00572          END IF
00573          ANG10 = PI/2.0
00574          PSI10 = 90.0
00575          SINPSI10 = 1.0
00576          COSPSI10 = 0.0
00577          28 CONTINUE

```

```

00578
C ***** LOW ETA REGION. FIT EXPONENTIAL AT PSICRPL FOR BEYOND PEAK.
C ***** IF PSICRPL < 90 CAN'T BE OBTAINED, USE PSICRPL = 90.
00581      ANGFRAD = 90.0
00582      ETACRPL = 0.5*BETA1
00583      SINCRPL = 1.0/SQRT(4.0*ETA**2+FMACH1**2)
00584      IF(ETA.GT.ETACRPL) ANGFRAD=ASIN(SINCRPL)*180./PI
00585      PSICRPL = ANGFRAD
00586      ANGFRAD = ANGFRAD*PI/180.0
00587      SINFRAD = SIN(ANGFRAD)
00588      ARG = PI*ETA*SINFRAD/SQRT(1.0-FMSQ1*SINFRAD**2)
00589      SINARG = SIN(ARG)
00590      PSQRATPL = (SINARG/ARG)**2
00591      ACPL = ALOG(PSQRATPL)/ANGFRAD**2
00592 C ***** ACPL IS THE AMPLITUDE OF THE EXPONENTIAL AT PSICRPL *****
00593 C ***** END LOW ETA REGION EXPONENTIAL FIT AT PSICRPL. *****
00594
00595 C CALCULATE P**2 AMPLITUDE CHANGE DUE TO ANGLE SHIFT FROM PSI = 0 TO
00596 C PSI = PSIPK2 FOR THE PLANE WAVE PASSING THROUGH THE JET SHEAR LAYER
00597
00598      PSQPKMUL = 1.0
00599      AREA1 = 1.0-COSPSI10
00600      AREA2 = 1.0+SINPSI2*SINPSI10-COSPSI2*COSPSI10
00601
00602      IF(PSIPK2.GT.0.0.AND.PSI10.LT.PSIPK2) THEN
00603      AREA2 = 2.0*SINPSI2*SINPSI10
00604      END IF
00605
00606      PSQPKMUL = AREA1/AREA2
00607      PSQPK = PSQPK*PSQPKMUL
00608
00609      CKPSI0 = -PSI10
00610      SUMPSQ = 0.0
00611
00612
00613      DO 40 I=1,NANGLE
00614      FI = I
00615      ANGDEG2 = ANGLE(I)
00616      IF(ANGDEG2.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00617      PSQRADT(I) = 0.0
00618      PSQTOT(I) = 0.0
00619      PSQTLOS(I) = 0.0
00620      GO TO 40
00621      END IF
00622 C
00623      ANGRAD2 = ANGDEG2*PI/180.0
00624      ANGDEG1 = ANGDEG2-DELPSE
00625      ANGRAD1 = ANGDEG1*PI/180.0
00626
00627      IF(ANGDEG1.LT.CKPSI0) GO TO 40
00628
00629      SINANG = SIN(ANGRAD1)
00630      COSANG = COS(ANGRAD1)
00631
00632      Q1DEN = SQRT(1.0-FMSQ1*SINANG**2)
00633      Q1 = SINANG/Q1DEN
00634      ARG = PI*ETA*Q1
00635      SINSQNUM = (SIN(ARG))**2
00636      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00637      PSQRAT = 1.0

```

```

00638
00639      PSQDEN = ARG**2
00640      IF(PSQDEN.LT.1.E-06.AND.ANGDEG1.LE.90.0) GO TO 49
00641
00642
00643      C ** LOW FREQUENCY REGION. FOLLOWING ALLOWS REASONABLE
      C ** EXTRAPOLATION BEYOND 90 deg, STARTING AT ARG = PI/2 IF ETA>ETAcrit
      C ** OR AT 90 deg. OTHERWISE. USES EXPONENTIAL FIT AT PSI = PSIPK1
00646      C ** AND PSI = PSIFIT.
00647
00648      IF(ANGDEG1.LT.PSICRPL) GO TO 48
00649
00650      QEXP = ACPL*(ANGDEG1)**2
00651      IF(QEXP.LT.-20.) QEXP=-20.
00652
00653      PSQRAT = EXP(QEXP)
00654      GO TO 49
00655
00656
00657      48 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00658      49 CONTINUE
00659
00660      PSQ = PSQRAT*PSQPK*GG
00661
00662      PSQRAD = PSQ
00663      C
00664      RAD = DISTANCE
00665      IF(ISIDELN.EQ.1) THEN
00666      RAD = DISTANCE/SIN(ANGDEG2*PI/180.0)
00667      END IF
00668
00669      PSQ = PSQ/RAD**2
00670
00671      C
00672      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00673      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00674      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCHF
00675
00676      C ***** NOTE THAT A TRANSMISSION LOSS (TLCHF) HAS BEEN USED *****
00677
00678      40 CONTINUE
00679      41 CONTINUE
00680      C
00681      70 CONTINUE
00682
00683      FNANGLE = NANGLE
00684      SUMWATT = 0.0
00685      DO 75 I=1,NANGLE
00686      ANGRAD = ANGLE(I)*PI/180.0
00687      SUMWATT = SUMWATT+PSQRADT(I)*SIN(ANGRAD)
00688
00689      C ** CONVERTING TO ANGLE FROM INLET AXIS FOR THIS AFT RADIATED NOISE
00690
00691      ANGLE(I) = 180.0-ANGLE(I)
00692      IF(PSQTOT(I).LT.4.E-08) THEN
00693      SPLTL(I) = 20.0
00694      SPL(I) = 20.0
00695      GO TO 75
00696      END IF
00697      SPLTL(I) = 10.0*ALOG10(PSQTLOS(I))+93.9794

```

```

00698      SPL(I) = 10.0*ALOG10(PSQTOT(I))+93.9794
00699      75 CONTINUE
00700
00701      WATTINT = SUMWATT*PI/(POWCON*FNANGLE)
00702      SPLDIF = 10.0*ALOG10(WATTS/WATTINT)
00703
00704      DO 80 I=1,NANGLE
00705      SPLTL(I) = SPLTL(I)+SPLDIF
00706      SPL(I) = SPL(I)+SPLDIF
00707      80 CONTINUE
00708
00709
00710      RETURN
00711      END

00001  C
00002  C *****
00003  C ***** END OF MAIN SUBROUTINE "BBRDCFEX" *****
00004  C ***** ALTERED 02/19/1998, E. J. RICE *****
00005  C *****
00006  C
00007  C
00008  C*****
00009  C ** SUBROUTINE FOR CALC NOZZLE CONDITIONS, FINAL JET MACH NUMBER,
      C VELOCITY, AND DIAMETER. ** NOTE ** OUTER DIAM CHANGES FOR NOZZLE.
      C
00012      SUBROUTINE CONOZ(TTOT,PTOT,PSTS,HTRAT,ANOZRAT,DDUCT,FMACHD,FMACH1
00013      1,CJET,DJET,TJET,PNOZ,DNOZ,CNOZ,FMACHN)
00014
00015  C
00016      PI = 3.1415927
00017      QM = 1.0+0.2*FMACHD**2
00018      DIN = DDUCT*HTRAT
00019      ADUCT = PI*(DDUCT**2-DIN**2)/4.0
00020      ANOZ = ADUCT*ANOZRAT
00021      DNOZ = SQRT(4.0*ANOZ/PI+DIN**2)
00022      TDUCT = TTOT/QM
00023      VSOND = 49.0421*SQRT(TDUCT)
00024      VDUCT = FMACHD*VSOND
00025      RHOT = 144.0*PTOT/(53.3*TTOT)
00026      RHOD = RHOT/QM**2.5
00027      FMASS = RHOD*ADUCT*VDUCT
00028  C ***** NOTE - AREAS ARE IN SQUARE INCHES *****
00029  C
00030  C ***** SOLVE FOR NOZZLE MACH NUMBER *****
00031  C
00032      QQDUCT = FMACHD/QM**3/ANOZRAT
00033      FN = 1.0
00034      DIFP = FN/(1.0+0.2*FN**2)**3-QQDUCT
00035      FNP = 1.0
00036
00037      DO 10 I=1,50
00038      FN = 0.975*FN
00039      DIF = FN/(1.0+0.2*FN**2)**3-QQDUCT
00040  C      WRITE(3,100) I,FN,DIF
00041  C 100 FORMAT(/,' I=',I2,' FN=',F7.4,' DIF=',1PE9.2)
00042      IF(DIF.LE.0.0) GO TO 12
00043      FNP = FN
00044      DIFP = DIF

```

```

00045      10 CONTINUE
00046          FN = QQDUCT
00047      111 DO 11 I=1,10
00048          FN = QQDUCT*(1.0+0.2*FN**2)**3
00049 C      WRITE(3,102) I,FN
00050 C 102 FORMAT(/,' REFINED, I=',I2,' FN=',F7.4)
00051
00052      11 CONTINUE
00053          GO TO 14
00054      12 CONTINUE
00055          FN = (DIFP*FN-DIF*FNP)/(DIFP-DIF)
00056 C      WRITE(3,101) FN
00057 C 101 FORMAT(/,' INTERPOLATED FN =',F7.4)
00058
00059          GO TO 111
00060      14 CONTINUE
00061          FMACHN = FN
00062 C      WRITE(3,103) FMACHN
00063 C 103 FORMAT(/,' FINAL ***** FMACHN =',F7.4)
00064
00065 C ***** "FMACH1" FINAL JET VELOCITY *****
00066
00067          FMACH1 = SQRT(5.0*((PTOT/PSTS)**(2./7.)-1.0))
00068          QNTJ = 1.0+0.2*FMACH1**2
00069          TJET = TTOT/QNTJ
00070          RHOJ = RHOT/QNTJ**2.5
00071          CJET = 49.0421*SQRT(TJET)
00072          VJET = CJET*FMACH1
00073          AJET = FMACH1/(RHOJ*VJET)
00074          DJET = SQRT(4.0*AJET/PI+DIN**2)
00075
00076 C ** FOLLOWING NOT USED HERE BUT MIGHT BE HANDY FOR LATER USE *****
00077 C ** CHECK NOZZLE THROAT PRESSURE VRS. SURROUNDING AMBIENT PRESSURE **
00078          QMN = 1.0/(1.0+0.2*FMACHN**2)
00079          PNOZ = PTOT*QMN**3.5
00080          TNOZ = TTOT*QMN
00081          CNOZ = 49.0421*SQRT(TNOZ)
00082          RETURN
00083          END
00001 C
00002 C *****
00003 C ***** END OF SUBROUTINE "CONOZ" *****
00004 C *****
00005

```

2.2.2.7 Sample Run Program ROTOR

Current code name is RDIRNEW1. To run the program first move to the directory that the code executable file resides in, then invoke the program by typing after the system prompt. From a Unix system type "program_name." From a VAX system type "run program_name." A typical run will look like this:

```

prompt> rdirnew1
Enter input file name : bbn_input.dat
Enter output file name : rotor_power.dat
Enter 2nd output file name: rotor_debug.dat
Enter spl plot output file name: rotor_spl.dat
prompt>

```

2.2.2.8 Sample Input File

```
$Input
RPM = 9782.2
RHO = 0.079005
DTIP = 22
HTR = 0.43
NBLADE = 22
NSTR = 12
NVANE = 54
GAM = 1.4
KASE = 1
LINLET = 0.99
LEXIT = 1.99
IABSOR = 0
NBSTD = 22
SCLOPTR = 4
SCLOPTS = 2
NHM = 10
BW = 0
NF = 11
NTOBNI = 7
NCOF = 15
RADMIC = 20
ISIDELN = 1
ALIP = 5.558
BLIP = 1.05
MACHS = 0.2
ANOZRAT = 0.6875
ETAFAN = 0.920
DELANG = 10.0
ITL = 1
TOBN = 36 37 38 39 40 41 42 43 44 45 46
SEMA = 0.418,0.422,0.428,0.436,0.444,0.450,0.453,0.454,0.451,
0.447,0.443,0.438,
SPERC = 4.77, 5.23, 5.77, 6.44, 7.32, 8.54,10.32, 9.00, 9.41,
.63, 6.77, 6.77,
SINCDR = 11.84,20.04,24.57,29.37,32.30,32.08,34.45,35.49,34.85,
32.70,30.83,27.32,
SATIR = 0.095,0.002,0.001,0.001,0.001,0.001,0.001,0.001,0.001,
.001,0.001,0.001,
SATIS = 0.081,0.021,0.018,0.021,0.023,0.023,0.022,0.018,0.016,
.013,0.012,0.024,
SELINR = 12*0.110,
SELINS = 0.128,0.063,0.064,0.067,0.072,0.075,0.092,0.091,0.126,
0.132,0.119,0.106,
SEMT = 0.81608 0.78311 0.74889 0.71207 0.67321 0.63208
.58778 0.53878 0.50051 0.46351 0.43564 0.41233
SCO = 1138.1 1132.5 1130.2 1129.9 1129.2 1127.8
.1 1124.4 1122.8 1121.3 1120.3 1119
STHETA = 0.28304 0.29918 0.32337 0.35848 0.39996 0.44971
.51579 0.61069 0.70143 0.81039 0.90401 0.99027
SAXSP = 1.4713 1.4837 1.5001 1.5247 1.5563 1.5928
.6373 1.6972 1.7526 1.8172 1.8753 1.9333
SROTCD = 0.06511 0.036664 0.016138 0.01422 0.012293 0.0066824
.0019007 0.0026553 0.0013424 0.0013326 0.0069932 0.01341
SSADIN = 12.289 11.351 11.008 11.028 11.187 11.454
.877 12.526 13.033 13.539 13.868 14.172
SCHDR = 4.227 4.124 4.020 3.911 3.794 3.675
.546 3.399 3.285 3.176 3.096 3.028
SCHDS = 1.839 1.886 1.921 1.947 1.964 1.970
```

```

          .964    1.944    1.919    1.888    1.861    1.837
STPRIN =    1.344    1.363    1.365    1.355    1.339    1.321
          .299    1.271    1.252    1.235    1.223    1.214
SDIA = 21.988 20.939 19.844 18.699 17.494 16.214
          .829   13.298   12.101   10.962   10.126   9.440
SSTATCD =    0.02     0.02     0.02     0.02     0.02     0.02
          .02     0.02     0.02     0.02     0.02     0.02
SINCD5 =    1  1    1  1  1    1  1  1    1  1  1    1  1
SATIW =     0.1     0.1     0.1     0.1     0.1     0.1
          .1      0.1     0.1     0.1     0.1     0.1
SCONTR =    1  1    1  1  1    1  1  1    1  1  1    1  1
SCONTS =    1  1    1  1  1    1  1  1    1  1  1    1  1
SCONTW =    1  1    1  1  1    1  1  1    1  1  1    1  1
SELINW =    0.1     0.1     0.1     0.1     0.1     0.1
          .1      0.1     0.1     0.1     0.1     0.1
SSCLR =    1  1    1  1  1    1  1  1    1  1  1    1  1
SSCLS =    1  1    1  1  1    1  1  1    1  1  1    1  1
SSCLW =    1  1    1  1  1    1  1  1    1  1  1    1  1
STVELR =    1  1    1  1  1    1  1  1    1  1  1    1  1
STVELS =    1  1    1  1  1    1  1  1    1  1  1    1  1
STVELW =    1  1    1  1  1    1  1  1    1  1  1    1  1
$

```

2.2.2.9 Sample Output Files

The ROTOR power and directivity output files are shown in the following sections. The debug output file is not listed.

2.2.2.9.1 Power Output File from ROTOR

```

HARD WALL ASSUMED
MICROPHONE IS ON A SIDELINE
MICROPHONE DISTANCE IN FEET IS =      20.00000
MACH NUMBER OF SURROUNDING MEDIUM =    0.2000000
ESTIMATED FAN ADIABATIC EFFICIENCY =    0.9200000

PROGRAM *** RDIRNEW1 ***

RESPONSE OF AN ISOLATED ROTOR
TO INGESTION OF INLET TURBULENCE

CASE NUMBER    1    OF    1

***** STRIP AREA NUMBER    1

      EMA      EMTIP      TI      SINCD      CONTR      L/SSTD
0.418      0.816      0.0950      11.84      1.000      0.11

      GAM      RHO      C      SDIA      SPERC      TPR
1.400      0.0790      1138.      21.988      4.770      1.344

      RPM      NB      NBSTD      HTR      DTIP      CHDR
9782.2      22      22      0.430      22.000      4.227

      EMR      RSCAL      RVEL      ELT      TIT      AR
0.900      1.00      1.0000      0.110      0.0950      1.483

CDROTOR =      6.5109998E-02
INLET LENGTH/TIP DIAMETER =      0.9900000
STHI,STHUSED      0.2830400      0.2830400

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

```

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	114.4	116.0	118.1	116.2	119.6	119.1	122.2
5011	1.3973	114.1	115.4	123.5	121.8	124.0	122.7	121.8
6309	1.7591	115.0	116.3	124.1	123.6	124.6	124.3	120.8
7943	2.2146	118.0	121.6	124.2	122.3	125.2	125.0	119.8
10000	2.7880	111.5	114.1	126.2	124.6	126.4	125.0	118.3
12589	3.5099	110.7	113.3	126.7	125.1	126.8	125.3	115.6
15848	4.4187	111.3	114.7	126.8	125.3	126.9	125.6	113.0
19952	5.5628	107.2	111.6	126.4	124.9	126.5	125.1	110.3
25118	7.0031	103.5	108.9	124.5	123.0	124.5	123.2	107.6
31622	8.8164	99.6	105.9	120.9	120.6	121.0	120.8	105.0
39810	11.0992	95.7	102.2	118.4	118.8	118.4	118.9	102.3

***** STRIP AREA NUMBER 2

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.422	0.783	0.0020	20.04	1.000	0.11

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1133.	20.939	5.230	1.363

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	4.124

EMR	RSCAL	RVEL	ELT	TIT	AR
0.872	1.00	1.0000	0.110	0.0020	1.520

CDROTOR = 3.6664002E-02
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.2991800 0.2991800

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	78.6	81.6	88.4	86.9	88.9	88.1	118.8
5011	1.3973	79.5	81.0	91.8	90.3	92.1	90.8	118.9
6309	1.7591	81.5	82.0	93.9	93.4	94.1	93.7	118.9
7943	2.2146	82.6	84.4	95.7	93.8	95.9	94.3	118.9
10000	2.7880	79.2	81.8	96.0	94.2	96.1	94.5	118.0
12589	3.5099	77.4	81.4	97.2	95.7	97.2	95.9	117.0
15848	4.4187	76.0	79.3	96.5	95.3	96.6	95.4	116.0
19952	5.5628	72.4	76.3	96.0	94.7	96.0	94.8	113.5
25118	7.0031	69.7	75.4	94.3	93.7	94.3	93.7	110.8
31622	8.8164	66.2	71.8	90.4	90.9	90.5	90.9	108.2
39810	11.0992	61.9	68.0	88.2	89.1	88.2	89.1	105.5

***** STRIP AREA NUMBER 3

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.428	0.749	0.0010	24.57	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1130.	19.844	5.770	1.365
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	4.020
EMR	RSCAL	RVEL	ELT	TIT	AR
0.844	1.00	1.0000	0.110	0.0010	1.560

CDROTOR = 1.6138000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.3233700 0.3233700

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	MUGRIDGE
3981	1.1099	74.9	76.6	86.3	84.6	86.6	85.2	110.9
5011	1.3973	74.7	76.5	87.8	86.1	88.0	86.5	111.9
6309	1.7591	73.6	75.5	88.2	88.1	88.3	88.3	112.9
7943	2.2146	72.5	74.9	90.0	88.6	90.0	88.7	113.9
10000	2.7880	72.2	74.9	92.3	90.8	92.3	90.9	114.5
12589	3.5099	72.2	75.6	92.3	91.2	92.4	91.3	114.5
15848	4.4187	70.6	74.7	91.4	90.7	91.5	90.8	114.5
19952	5.5628	65.6	70.6	91.4	90.6	91.4	90.7	114.1
25118	7.0031	62.5	68.1	89.1	88.8	89.1	88.8	113.1
31622	8.8164	59.4	65.2	85.9	86.8	86.0	86.9	112.1
39810	11.0992	55.4	62.2	83.7	85.6	83.7	85.6	110.5

***** STRIP AREA NUMBER 4

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.436	0.712	0.0010	29.37	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1130.	18.699	6.440	1.355
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.911
EMR	RSCAL	RVEL	ELT	TIT	AR
0.815	1.00	1.0000	0.110	0.0010	1.603

CDROTOR = 1.4220000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.3584800 0.3584800

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	MUGRIDGE
3981	1.1099	74.5	77.2	89.3	87.5	89.4	87.9	109.1
5011	1.3973	74.6	77.8	91.7	89.8	91.8	90.1	110.1
6309	1.7591	76.0	77.8	92.3	91.8	92.4	92.0	111.1
7943	2.2146	73.8	76.5	92.1	91.5	92.2	91.7	112.1
10000	2.7880	74.8	78.5	91.9	91.0	92.0	91.3	113.1
12589	3.5099	70.0	74.2	92.5	92.1	92.5	92.2	113.1
15848	4.4187	67.4	72.1	93.7	92.9	93.7	92.9	113.1
19952	5.5628	65.3	70.7	93.6	92.9	93.6	92.9	113.1
25118	7.0031	62.6	68.5	91.0	91.1	91.0	91.1	112.2
31622	8.8164	59.1	65.4	88.4	89.3	88.4	89.3	111.2
39810	11.0992	54.9	61.7	85.4	87.1	85.4	87.1	110.2

***** STRIP AREA NUMBER 5

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.444	0.673	0.0010	32.30	1.000	0.11

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1129.	17.494	7.320	1.339

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.794

EMR	RSCAL	RVEL	ELT	TIT	AR
0.786	1.00	1.0000	0.110	0.0010	1.653

CDROTOR = 1.2293000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.3999600 0.3999600

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	MUGRIDGE
3981	1.1099	73.7	76.9	90.2	88.5	90.3	88.8	107.0
5011	1.3973	76.4	78.8	92.8	91.0	92.9	91.2	108.0
6309	1.7591	74.7	77.2	94.5	93.6	94.5	93.7	109.0
7943	2.2146	76.3	79.5	95.5	94.6	95.6	94.8	110.0
10000	2.7880	74.9	78.6	96.4	95.3	96.5	95.4	111.0
12589	3.5099	70.9	75.7	96.9	96.1	96.9	96.2	111.7
15848	4.4187	68.9	74.4	96.6	96.1	96.6	96.1	111.7
19952	5.5628	66.3	72.5	95.6	95.4	95.6	95.4	111.7
25118	7.0031	63.3	70.3	93.6	94.0	93.6	94.0	111.3
31622	8.8164	60.6	67.5	90.9	92.2	90.9	92.2	110.3
39810	11.0992	55.9	63.5	88.4	90.3	88.4	90.3	109.3

***** STRIP AREA NUMBER 6

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.450	0.632	0.0010	32.08	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1128.	16.214	8.540	1.321
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.675
EMR	RSCAL	RVEL	ELT	TIT	AR
0.754	1.00	1.0000	0.110	0.0010	1.706

CDROTOR = 6.6824001E-03
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.4497100 0.4497100

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	73.3	75.8	88.8	87.2	88.9	87.5	100.9
5011	1.3973	73.3	75.7	91.0	89.4	91.1	89.6	101.9
6309	1.7591	73.9	75.3	93.0	92.0	93.0	92.1	102.9
7943	2.2146	73.7	76.1	94.2	93.5	94.3	93.6	103.9
10000	2.7880	73.0	76.3	95.3	94.3	95.4	94.4	104.9
12589	3.5099	70.0	74.5	96.2	95.5	96.2	95.5	105.9
15848	4.4187	68.6	73.6	96.8	96.2	96.8	96.2	106.9
19952	5.5628	66.7	72.5	97.2	96.6	97.2	96.6	107.9
25118	7.0031	64.9	71.2	97.3	96.8	97.3	96.8	108.1
31622	8.8164	62.8	69.3	97.3	96.8	97.3	96.8	108.1
39810	11.0992	60.6	67.5	97.0	96.7	97.0	96.7	108.1

***** STRIP AREA NUMBER 7

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.453	0.588	0.0010	34.45	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1126.	14.829	10.320	1.299
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.546
EMR	RSCAL	RVEL	ELT	TIT	AR
0.718	1.00	1.0000	0.110	0.0010	1.768

CDROTOR = 1.9007000E-03
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.5157900 0.5157900

STREAMLINE LIFT COEFFICIENT CALCULATED USING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	71.3	73.5	86.9	85.1	87.0	85.4	89.0
5011	1.3973	72.1	74.3	88.9	87.5	89.0	87.7	90.0
6309	1.7591	72.2	72.9	90.7	89.7	90.8	89.8	91.0
7943	2.2146	71.8	73.4	91.8	91.2	91.9	91.3	92.0
10000	2.7880	70.6	73.2	92.6	91.9	92.7	92.0	93.0
12589	3.5099	68.1	72.2	93.2	92.7	93.2	92.8	94.0
15848	4.4187	65.9	70.6	93.3	93.1	93.3	93.2	95.0
19952	5.5628	63.6	68.9	93.2	93.3	93.3	93.3	96.0
25118	7.0031	61.6	67.3	93.0	93.2	93.0	93.2	97.0
31622	8.8164	59.4	65.2	92.6	92.9	92.6	92.9	98.0
39810	11.0992	56.9	63.0	92.1	92.4	92.1	92.4	99.0

***** STRIP AREA NUMBER 8

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.454	0.539	0.0010	35.49	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1124.	13.298	9.000	1.271
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.399
EMR	RSCAL	RVEL	ELT	TIT	AR
0.686	1.00	1.0000	0.110	0.0010	1.845

CDROTOR = 2.6553001E-03
 INLET LENGTH/TIP DIAMETER = 0.9900000
 STHI,STHUSED 0.6106900 0.6106900

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	66.5	69.0	82.1	80.2	82.2	80.5	89.6
5011	1.3973	67.3	69.1	84.0	82.4	84.0	82.6	90.6
6309	1.7591	67.5	68.0	85.9	84.8	85.9	84.9	91.6
7943	2.2146	68.0	69.4	87.1	86.3	87.1	86.4	92.6
10000	2.7880	66.2	68.8	88.0	87.3	88.1	87.3	93.6
12589	3.5099	66.2	69.6	88.6	88.1	88.7	88.2	94.6
15848	4.4187	64.9	68.8	88.8	88.6	88.8	88.7	95.6
19952	5.5628	59.8	64.5	88.7	88.8	88.7	88.8	96.6
25118	7.0031	57.4	62.6	88.5	88.8	88.5	88.8	97.6
31622	8.8164	54.9	60.6	88.0	88.5	88.0	88.5	98.6
39810	11.0992	52.3	58.3	87.5	88.1	87.5	88.1	99.6

***** STRIP AREA NUMBER 9

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.451	0.501	0.0010	34.85	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1123.	12.101	9.410	1.252
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.285
EMR	RSCAL	RVEL	ELT	TIT	AR
0.656	1.00	1.0000	0.110	0.0010	1.909

CDROTOR = 1.3424000E-03
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.7014300 0.7014300

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	63.0	66.2	79.1	77.2	79.2	77.6	82.3
5011	1.3973	65.7	67.9	80.6	79.3	80.8	79.6	83.3
6309	1.7591	67.6	68.0	82.6	81.5	82.7	81.7	84.3
7943	2.2146	66.0	66.6	84.1	83.1	84.2	83.2	85.3
10000	2.7880	64.6	66.8	85.1	84.2	85.2	84.3	86.3
12589	3.5099	64.4	66.7	85.6	85.0	85.6	85.1	87.3
15848	4.4187	61.4	63.9	85.7	85.5	85.8	85.6	88.3
19952	5.5628	58.9	62.3	85.8	85.8	85.8	85.8	89.3
25118	7.0031	56.2	60.0	85.4	85.8	85.4	85.8	90.3
31622	8.8164	54.1	58.3	85.0	85.5	85.0	85.5	91.3
39810	11.0992	51.4	55.7	84.4	85.0	84.4	85.1	92.3

***** STRIP AREA NUMBER 10

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.447	0.464	0.0010	32.70	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1121.	10.962	7.630	1.235
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.176
EMR	RSCAL	RVEL	ELT	TIT	AR
0.631	1.00	1.0000	0.110	0.0010	1.974

CDROTOR = 1.3326000E-03
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.8103900 0.8103900

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	60.8	63.3	75.6	73.0	75.8	73.5	79.9
5011	1.3973	59.7	62.1	77.0	75.3	77.0	75.5	80.9
6309	1.7591	61.6	63.1	78.2	77.3	78.2	77.5	81.9
7943	2.2146	60.0	60.8	79.0	78.6	79.1	78.7	82.9
10000	2.7880	62.2	64.8	79.8	79.3	79.9	79.4	83.9
12589	3.5099	64.4	67.7	80.9	80.1	81.0	80.3	84.9
15848	4.4187	57.8	60.2	81.3	80.7	81.3	80.8	85.9
19952	5.5628	56.0	59.1	81.5	81.2	81.5	81.2	86.9
25118	7.0031	53.0	56.4	81.2	81.2	81.2	81.2	87.9
31622	8.8164	50.5	54.4	80.8	81.1	80.8	81.1	88.9
39810	11.0992	48.0	52.2	80.2	80.7	80.2	80.7	89.9

***** STRIP AREA NUMBER 11

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.443	0.436	0.0010	30.83	1.000	0.11

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1120.	10.126	6.770	1.223

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.096

EMR	RSCAL	RVEL	ELT	TIT	AR
0.611	1.00	1.0000	0.110	0.0010	2.025

CDROTOR = 6.9932002E-03
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.9040100 0.9040100

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	64.9	67.1	73.2	70.3	73.8	72.0	92.6
5011	1.3973	61.0	63.0	75.0	72.8	75.2	73.2	93.6
6309	1.7591	63.2	64.9	76.7	75.1	76.9	75.5	94.6
7943	2.2146	65.8	68.9	78.1	77.0	78.4	77.6	95.6
10000	2.7880	60.4	63.2	79.0	78.0	79.1	78.2	96.6
12589	3.5099	58.1	60.7	79.4	78.7	79.5	78.8	97.6
15848	4.4187	55.6	58.7	79.4	79.2	79.5	79.2	98.6
19952	5.5628	53.1	56.9	79.2	79.3	79.2	79.3	99.0
25118	7.0031	52.3	56.6	78.8	79.1	78.8	79.1	99.0
31622	8.8164	48.1	52.3	78.2	78.7	78.2	78.7	99.0
39810	11.0992	45.2	50.0	77.5	78.1	77.5	78.2	98.4

***** STRIP AREA NUMBER 12

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.438	0.412	0.0010	27.32	1.000	0.11

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1119.	9.440	6.770	1.214

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.028

EMR	RSCAL	RVEL	ELT	TIT	AR
0.592	1.00	1.0000	0.110	0.0010	2.071

CDROTOR = 1.3410000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000
STHI,STHUSED 0.9902700 0.9902700

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 4

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL		MUGRIDGE
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST	
3981	1.1099	61.7	63.5	71.0	68.1	71.4	69.4	97.1
5011	1.3973	58.8	60.7	72.9	70.7	73.1	71.1	98.1
6309	1.7591	60.6	61.6	74.6	73.0	74.7	73.3	99.1
7943	2.2146	61.3	62.8	76.0	74.9	76.1	75.2	100.1
10000	2.7880	58.3	61.2	77.0	76.1	77.1	76.2	100.6
12589	3.5099	56.5	59.2	77.4	76.8	77.5	76.8	100.6
15848	4.4187	54.0	57.3	77.4	77.2	77.5	77.3	100.6
19952	5.5628	51.8	55.8	77.3	77.4	77.3	77.4	100.0
25118	7.0031	49.3	53.7	76.8	77.1	76.8	77.1	99.0
31622	8.8164	46.8	51.4	76.2	76.7	76.2	76.7	98.0
39810	11.0992	43.7	49.0	75.5	76.2	75.5	76.2	96.2

FAN TOTAL POWER SPECTRUM

TOBN	F/BPF	PWL-UP	PWL-DN	PWL-TOT	MUGR-TOT
36	1.1099	119.66	119.13	122.41	127.30
37	1.3973	123.98	122.70	126.40	127.23
38	1.7591	124.60	124.34	127.48	126.85
39	2.2146	125.18	125.00	128.11	126.60
40	2.7880	126.39	125.02	128.77	125.93
41	3.5099	126.85	125.37	129.18	124.94
42	4.4187	126.95	125.65	129.36	124.16
43	5.5628	126.52	125.17	128.90	123.13
44	7.0031	124.52	123.23	126.93	121.93
45	8.8164	121.01	120.83	123.93	120.76
46	11.0992	118.46	118.93	121.71	119.55

2.2.2.9.2 Directivity Output File from ROTOR

ROTOR SPL PLOT OUTPUT FILE

FREQUENCY = 3981, OBN = 36

ANGLE INL	SPL EXH	SPL TOT	SPL
10.0	64.0	40.9	64.0
20.0	65.0	51.9	65.2
30.0	80.8	59.7	80.8
40.0	79.4	65.9	79.6
50.0	82.5	71.1	82.8
60.0	86.5	75.4	86.8
70.0	88.3	78.9	88.8
80.0	83.5	84.2	86.9
90.0	73.7	81.1	81.8
100.0	67.9	80.8	81.0
110.0	63.2	82.4	82.5
120.0	58.2	82.2	82.2
130.0	52.6	81.4	81.4
140.0	46.1	86.6	86.6
150.0	37.9	60.3	60.3
160.0	26.8	19.7	27.6
170.0	20.5	19.7	23.2

FREQUENCY = 5011, OBN = 37

ANGLE INL	SPL EXH	SPL TOT	SPL
10.0	57.8	46.7	58.1
20.0	66.9	57.7	67.4
30.0	82.3	65.5	82.4
40.0	82.8	71.7	83.1
50.0	86.8	76.9	87.2
60.0	91.4	81.2	91.8
70.0	93.8	84.8	94.3
80.0	87.5	90.1	92.0
90.0	76.1	86.3	86.7
100.0	71.1	84.9	85.0
110.0	66.5	84.3	84.4
120.0	61.6	82.4	82.4
130.0	56.2	80.8	80.9
140.0	49.8	89.7	89.7
150.0	41.7	57.3	57.4
160.0	30.7	19.8	31.0
170.0	20.5	19.8	23.2

FREQUENCY = 6309, OBN = 38

ANGLE INL	SPL EXH	SPL TOT	SPL
10.0	54.5	47.8	55.4
20.0	65.4	58.9	66.3
30.0	78.7	66.8	78.9
40.0	83.1	73.1	83.5
50.0	88.7	78.4	89.1
60.0	92.5	82.7	93.0
70.0	94.4	86.4	95.0
80.0	80.4	91.5	91.8
90.0	82.7	89.1	90.0
100.0	77.3	87.3	87.7
110.0	72.3	88.7	88.8
120.0	67.2	83.7	83.8
130.0	61.5	77.2	77.3
140.0	54.9	90.5	90.5
150.0	46.7	58.0	58.3
160.0	35.6	19.9	35.7
170.0	20.2	19.9	23.1

FREQUENCY = 7943, OBN = 39

ANGLE INL	SPL EXH	SPL TOT	SPL
10.0	54.4	48.5	55.4
20.0	67.0	59.6	67.7
30.0	83.1	67.4	83.2
40.0	85.1	73.7	85.4
50.0	89.3	78.9	89.7
60.0	92.0	83.3	92.5
70.0	93.7	86.9	94.6
80.0	87.4	92.3	93.5
90.0	80.0	87.0	87.8
100.0	74.6	86.4	86.6
110.0	69.8	89.0	89.0
120.0	64.8	84.6	84.6
130.0	59.4	74.8	75.0
140.0	53.0	88.8	88.8
150.0	45.0	54.7	55.1
160.0	34.0	20.0	34.2
170.0	19.5	20.0	22.7

FREQUENCY = 10000, OBN = 40

ANGLE INL	SPL EXH	SPL TOT	SPL
10.0	49.3	49.4	52.4
20.0	61.7	60.5	64.1
30.0	85.6	68.3	85.7
40.0	84.5	74.6	84.9
50.0	88.8	79.8	89.3
60.0	94.1	84.2	94.5
70.0	96.3	87.9	96.9
80.0	87.9	92.9	94.1
90.0	80.2	86.8	87.7
100.0	75.0	84.5	84.9
110.0	70.4	86.2	86.3
120.0	65.7	85.0	85.1
130.0	60.4	76.8	76.9
140.0	54.1	92.6	92.6
150.0	46.2	57.7	58.0
160.0	35.3	19.7	35.4
170.0	19.0	19.7	22.3

FREQUENCY = 12589, OBN = 41

ANGLE INL	SPL EXH	SPL TOT	SPL
10.0	54.5	48.7	55.5
20.0	62.8	59.8	64.6
30.0	84.0	67.6	84.1
40.0	79.2	73.9	80.4
50.0	89.4	79.2	89.8
60.0	95.5	83.6	95.8
70.0	96.7	87.3	97.1
80.0	86.7	91.8	93.0
90.0	79.2	87.9	88.5
100.0	74.4	86.1	86.4
110.0	70.1	86.7	86.8
120.0	65.6	84.0	84.0
130.0	60.5	79.4	79.5
140.0	54.4	93.7	93.7
150.0	46.6	51.1	52.4
160.0	35.7	19.1	35.8
170.0	18.9	19.1	22.0

FREQUENCY = 15848, OBN = 42

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		53.0	48.6	54.4
20.0		60.7	59.7	63.2
30.0		80.5	67.5	80.7
40.0		83.5	73.8	83.9
50.0		90.6	79.1	90.9
60.0		96.1	83.5	96.4
70.0		96.1	87.1	96.6
80.0		84.6	90.9	91.8
90.0		77.6	86.1	86.7
100.0		73.3	85.0	85.3
110.0		69.3	84.5	84.7
120.0		65.0	82.3	82.4
130.0		60.0	79.6	79.7
140.0		54.0	95.1	95.1
150.0		46.3	50.3	51.7
160.0		35.5	18.3	35.6
170.0		18.3	18.3	21.3

FREQUENCY = 19952, OBN = 43

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		49.3	46.5	51.1
20.0		62.8	57.5	63.9
30.0		78.4	65.4	78.6
40.0		84.1	71.7	84.3
50.0		89.3	77.0	89.5
60.0		96.6	81.4	96.7
70.0		94.8	85.1	95.2
80.0		82.6	88.1	89.2
90.0		76.3	84.5	85.1
100.0		72.4	83.5	83.9
110.0		68.7	84.6	84.7
120.0		64.6	82.6	82.6
130.0		59.7	79.5	79.5
140.0		53.7	95.3	95.3
150.0		46.1	48.6	50.5
160.0		35.4	17.1	35.4
170.0		18.1	17.1	20.7

FREQUENCY = 25118, OBN = 44

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		47.0	42.6	48.3
20.0		60.6	53.7	61.4
30.0		69.6	61.6	70.2
40.0		80.9	67.9	81.1
50.0		88.4	73.2	88.5
60.0		95.2	77.6	95.3
70.0		91.4	81.4	91.8
80.0		79.4	84.5	85.6
90.0		74.0	82.0	82.7
100.0		70.6	81.3	81.7
110.0		67.0	82.4	82.6
120.0		63.0	80.3	80.4
130.0		58.2	77.4	77.4
140.0		52.3	93.8	93.8
150.0		44.7	43.7	47.3
160.0		34.1	15.9	34.1
170.0		18.2	15.9	20.2

FREQUENCY = 31622, OBN = 45

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		46.0	34.6	46.3
20.0		56.8	45.8	57.1
30.0		75.0	53.7	75.0
40.0		80.9	60.1	80.9
50.0		86.2	65.4	86.2
60.0		92.0	69.9	92.0
70.0		83.0	73.7	83.4
80.0		73.9	76.8	78.6
90.0		70.0	76.0	77.0
100.0		67.1	76.4	76.9
110.0		63.8	78.0	78.2
120.0		59.9	76.8	76.9
130.0		55.2	71.6	71.7
140.0		49.4	92.0	92.0
150.0		41.8	37.2	43.1
160.0		31.1	13.4	31.2
170.0		18.5	13.4	19.7

FREQUENCY = 39810, OBN = 46

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		46.7	31.3	46.9
20.0		58.2	42.3	58.3
30.0		79.1	50.2	79.1
40.0		77.8	56.5	77.9
50.0		83.5	61.8	83.5
60.0		88.5	66.2	88.5
70.0		79.8	69.9	80.2
80.0		75.3	73.0	77.3
90.0		71.4	69.0	73.4
100.0		68.4	70.4	72.5
110.0		65.2	72.3	73.0
120.0		61.3	72.4	72.7
130.0		56.6	68.6	68.9
140.0		50.7	90.5	90.5
150.0		43.2	33.9	43.7
160.0		32.5	11.4	32.6
170.0		22.5	11.4	22.8

2.2.3 STATOR – Stator / Turbulence Interaction Noise Program

2.2.3.1 Description of Prediction Code

A discussion of the main program and subroutines of the prediction code is found in Section 2.2.2.1, page 4 (Description of Prediction Code, program ROTOR). The last paragraph highlights the differences between the ROTOR and STATOR codes.

2.2.3.2 Description of Input File

The input file is identical to the input file for program ROTOR. See Section 2.2.2.4, page 6.

2.2.3.3 Description of Output Files

The power output file and SPL directivity output file are the same format as those produced by program ROTOR except for header information. The files are described in Section 2.2.2.5, page 9.

2.2.3.4 STATOR Source Code Listing

```
00001  C      PROGRAM SDIRFIN.FOR
00002  C*#RUN *=;16130ER/PG/ROTIN2C(BCD,NOGO,CORE=2)
00003  C      ROTIN          STATOR/TURBULENCE INTERACTION NOISE PREDICTION
00004  C
00005      DIMENSION AEV(46),AED(46),ZMM(46),ZPP(46),DCV1(91),DCD1(91),
00006      &          F(200),DCV2(91),DCD2(91),STHOSR(91),AEVETC(2,91),
00007      &          AEDETC(2,91),TOBN(200),SAXSP(21),SCHDS(21),
00008      &          TROGV(91),F3D(4,200,91),CTFRAT(200,91),
00009      &          CTFRATN(20000),IPERM(20000),CTFRATO(20000),
00010      &          POWTOT(20000),SPIMPR(200),SPIMPI(200),
00011      &          SPIMPRE(200),SPIMPIE(200),FINT(200),SPIMPZ2(200),
00012      &          SPIMPZ2E(200),RICEV(4,200,91),RICED(4,200,91),
00013      &          REROT1(91),REROT2(91)
00014  C
00015      DIMENSION SAREA(21),SCLR(21),SEMA(21),SEMT(21),SSIGR(21),
00016      &          SSADIN(21),FOB(200),PVT(200),PDT(200),ANGLEO(100),
00017      &          SCONTR(21),SCONTS(21),STVELR(21),SSCLR(21),
00018      &          STPRIN(21),SCO(21),SROTC(21),SDIA(21),SNCD(2,91),
00019      &          SPERC(21),SINCDR(21),SCHDR(21),SCLS(21),SCONTW(21),
00020      &          SINCD(21),SSCLS(21),SSCLW(21),STVELS(21),STVELW(21)
00021  C
00022      DIMENSION DVP(21),DDP(21),QVP(21),QDP(21),CVP(21),CDP(21),
00023      &          SSIGS(21),SDELU(21),NDSTLB(21),SALD(21),STHD(21),
00024      &          SPHD(21),SDELP(21),TVP(21),TDP(21),FJJ(21)
00025  C
00026      DIMENSION SATIR(21),SATIS(21),SATIW(21),SELINR(21),SELINS(21),
00027      &          SELINW(21),SNCU(2,91),SSTATCD(21),STHETA(21)
00028  C
00029      DIMENSION FDVTH(3,200,91),FDDTH(3,200,91),FQVTH(3,200,91),
00030      &          FQDTH(3,200,91),SWWND(21),TRROT1(91),TRROT2(91)
00031      DIMENSION SDVTH(200,91),SDDTH(200,91),SQVTH(200,91),
00032      &          SQDTH(200,91),STVTH(200,91),STDTH(200,91)
00033      DIMENSION WSUMDV(3,200),WSUMDD(3,200),WSUMQV(3,200),
00034      &          WSUMQD(3,200),WSNDV(3,200),WSNDD(3,200),WSNQV(3,200),
00035      &          WSNQD(3,200),WSDV(200),WSDD(200),WSQV(200),
00036      &          WSTV(200),WSTD(200),ANGLE(100),WSUMIN(200),
00037      &          WSQD(200),WSNDV1(200),WSNDD1(200),WSNQV1(200),
00038      &          WSNQD1(200),SPLVDB(100),SPLDDB(100),SPLOUT(100),
```

```

00039      &          SPLVDBT(100,200),SPLDDBT(100,200),SPLDB(100),
00040      &          SPLDBT(100),WUP(200,200),WDN(200,200),SPLTL(100)
00041      REAL      NDVTH(200,91),NDDTH(200,91),NQVTH(200,91),
00042      &          NQDTH(200,91),MACHS
00043      REAL LEXIT,IXP,IXM,NBNP,NBNM,KYS,KTOTP,KTOTM,KR,LEXLOC
00044      REAL ID,LINLET,LSHOCK,LINLOC,LINOVD,LEXOVD
00045      COMPLEX CTU,CTV,CTRM,CTEXP,CGX,CGY,CSYM,CASYM,CDEN,CDEN1,CPART,
00046      &          IM
00047      REAL      FREQNCY
00048      INTEGER FHZ
00049      INTEGER SCLOPTS,SCLOPTR
00050      LOGICAL CHECK
00051  C
00052      CHARACTER *60 INFILE,OUTFILE,OUTFILE1,PLOTFILE
00053  C
00054  C
00055      NAMELIST/INPUT/
00056      &          BW, DTIP, GAM, HTR, IDIST, IPRINT, KASE, NBLADE,
00057      &          NBSTD, NDSTLB, NF, NHM, NSTR, NVANE, RHO, RPM, SALD,
00058      &          SCHDR,SCLOPTS,SCLOPTR,SCO, SCONTR, SDELP, SDELU,
00059      &          SDIA, SELINR, SEMA, SEMT, SINCDR, SPERC, SPHD, SROTC,
00060      &          SSCLR, SIGS, STHD, STHETA, STPRIN, STVELR, TOBN,
00061      &          SATIR,SATIW,SATIS,SCLR,SCLS,SCONTS,SCONTW,SELINS,
00062      &          SELINW,SINCD,SSCLS,SSCLW,STVELS,STVELW,SCHDS,SAXSP,
00063      &          SSADIN,SSTATCD,LEXIT,LINLET,SPIMPR,SPIMPI,
00064      &          SPIMPRE,SPIMPIE,IABSOR,NTOBNI,NCOF,RADMIC,ISIDELN,
00065      &          ALIP,BLIP,MACHS,ANOZRAT,ETAFAN,DELANG,ITL
00066  C      OPEN INPUT FILE
00067  C
00068      WRITE(*,41)
00069      41 FORMAT(' Enter input file name : ',%)
00070      READ(*,42) INFILE
00071      42 FORMAT(A60)
00072  C
00073      OPEN (UNIT=11,STATUS='OLD',FORM='FORMATTED',FILE=INFILE,
00074      & ERR=45)
00075      GO TO 46
00076      45      PRINT *, 'INPUT FILE NOT FOUND'
00077      GO TO 10001
00078  C
00079  C      OPEN OUTPUT FILES
00080  C
00081      46 WRITE(*,47)
00082      47 FORMAT(' Enter output file name : ',%)
00083      READ(*,42) OUTFILE
00084  C
00085      OPEN (UNIT=12,STATUS='NEW',FORM='FORMATTED',FILE=OUTFILE,
00086      & CARRIAGECONTROL='LIST')
00087  C
00088      WRITE(*,48)
00089      48 FORMAT(' Enter 2nd output file name : ',%)
00090      READ(*,42) OUTFILE1
00091  C
00092      OPEN (UNIT=13,STATUS='NEW',FORM='FORMATTED',FILE=OUTFILE1,
00093      & CARRIAGECONTROL='LIST')
00094  C
00095      WRITE(*,49)
00096      49 FORMAT(' Enter spl plot output file name : ',%)
00097      READ(*,42) PLOTFILE
00098  C

```

```

00099      OPEN (UNIT=14,STATUS='NEW',FORM='FORMATTED',FILE=PLOTFILE,
00100      & CARRIAGECONTROL='LIST')
00101      C
00102      C      WRITE TO SPL PLOT OUTPUT FILE TO INDICATE STATOR INFO
00103      C
00104      WRITE(14,(''STATOR SPL PLOT OUTPUT FILE''))
00105      C
00106      C      READ INPUT DATA
00107      C
00108      KASE=1
00109      NCASE=0
00110      BW=0.0
00111      IM = CMPLX(0.0,1.0)
00112      C
00113      C****      LOOP TO PROCESS ALL CASES
00114      C
00115      DO WHILE (NCASE .LT. KASE)
00116      1      NCASE=NCASE+1
00117      READ(11,INPUT,ERR=1000,END=9999)
00118      IF(IABSOR.EQ.0) WRITE(12,*)'      HARD WALL ASSUMED'
00119      IF(IABSOR.NE.0) WRITE(12,*)'      TREATED WALL ASSUMED'
00120      FCOF = NCOF
00121      IF(ISIDELN.EQ.0)WRITE(12,*)'MICROPHONE IS ON AN ARC'
00122      IF(ISIDELN.EQ.1)WRITE(12,*)'MICROPHONE IS ON A SIDELINE'
00123      WRITE(12,*)' MICROPHONE DISTANCE IN FEET IS = ',RADMIC
00124      WRITE(12,*)' MACH NUMBER OF SURROUNDING MEDIUM = ',MACHS
00125      WRITE(13,*)'NOZZLE EXIT AREA/DUCT AREA = ',ANOZRAT
00126      WRITE(12,*)'ESTIMATED FAN ADIABATIC EFFICIENCY = ',ETAFAN
00127      IF ( ITL.NE.0 ) WRITE(13,*)' NO DUCT TRANS LOSS ASSUMED'
00128      IF ( ITL.NE.0 ) WRITE(13,*)' '
00129      WRITE(12,*)' '
00130      WRITE(12,108)
00131      WRITE(12,106) NCASE,KASE
00132      NJJ=10
00133      C
00134      CHECK = .TRUE.
00135      PI=3.1415926
00136      BPF=RPM*NBLADE/60.
00137      DSTAT=FLOAT(NBLADE)/FLOAT(NVANE)
00138      DO 20 J=1,NF
00139      DO 18 I = 1,NCOF
00140      WUP(I,J) = 0.00
00141      WDN(I,J) = 0.00
00142      18      CONTINUE
00143      DO 19 I = 1,100
00144      SPLVDBT(I,J) = -150.
00145      SPLDDBT(I,J) = -150.
00146      IF ( J.EQ.1 ) SPLDBT(I) = -150.
00147      19      CONTINUE
00148      PDT(J)=-150.
00149      PVT(J)=-150.
00150      FOB(J)=10.**(TOBN(J)*0.1)
00151      FINT(J)=FOB(J)/BPF
00152      SPIMPZ2(J) = SPIMPR(J)**2+SPIMPI(J)**2
00153      SPIMPZ2E(J)= SPIMPRE(J)**2+SPIMPIE(J)**2
00154      20      END DO
00155      C
00156      C      INDEX OVER STRIP NUMBER - KJI
00157      C
00158      IF ( NSTR.EQ.1 ) MIDSTR = 1

```

```

00159      IF ( NSTR.GT.1 ) MIDSTR = NSTR/2
00160      DO 1949 KJI=1,NSTR
00161      LINLOC = LINLET*DTIP/SDIA(KJI)
00162      LINOVD = FLOAT(NBLADE)*LINLOC/PI
00163      LEXLOC = LEXIT*DTIP/SDIA(KJI)
00164      LEXOVD = FLOAT(NBLADE)*LEXLOC/PI
00165      AR = 0.5*DTIP*(1.-HTR)/SCHDS(KJI)
00166      SAR = AR
00167      PER = SPERC(KJI)
00168      OD = SDIA(KJI)
00169      ID = (1.-0.01*PER)*OD
00170      RUPP = OD/DTIP
00171      IF ( RUPP.GT.1.00 ) RUPP = 1.00
00172      RLOW = RUPP*(1.-0.01*PER)
00173      IF ( RLOW.LT.HTR ) RLOW = HTR
00174      IF ( RLOW.LE.HTR ) RUPP = RLOW/(1.-0.01*PER)
00175      BNEW = DTIP*FLOAT(NBLADE)/(2.*PI*SDIA(KJI))
00176      ANEW = BNEW*HTR
00177      C      PRINT *, 'BNEW,ANEW ',BNEW,ANEW
00178      A = PI*(OD**2-ID**2)/576.
00179      CDR=SROTC(D(KJI))
00180      TI=SATIS(KJI)
00181      CONTR=SCONTS(KJI)
00182      SIGS=SCHDS(KJI)*FLOAT(NVANE)/(PI*SDIA(KJI))
00183      EMA=SEMA(KJI)
00184      EMT=SEMT(KJI)*(1.-0.005*PER)
00185      EMR=EMA
00186      EMRSP2=EMA**2
00187      EMRSP=SQRT(EMRSP2)
00188      SSAD=SSADIN(KJI)
00189      C Does this (the following line) make all that much sense ? Who knows!
00190      EL=SELINS(KJI)*FLOAT(NBLADE)/FLOAT(NBSTD)
00191      RSCAL=SSCLS(KJI)
00192      RVEL=STVELS(KJI)
00193      TPR=STPRIN(KJI)
00194      ELT=EL/RSCAL
00195      TIT=TI*RVEL
00196      AAA=A
00197      C=SCO(KJI)
00198      WWND=1.00
00199      SPCON=(EMA*C)*(EMRSP2)*(C**2)
00200      TABS = (C/49.0422)**2
00201      PABS = 53.3*TABS*RHO/144.
00202      IF (KJI.EQ.1) PTOT = PABS*TPR
00203      C
00204      C
00205      C      WRITE INPUT DATA
00206      C
00207      WRITE(12,118) KJI,WWND
00208      WRITE(12,120)
00209      WRITE(12,122) EMA,SEMT(KJI),TI,SINCDS(KJI),CONTR,
00210      &      SELINS(KJI)
00211      WRITE(12,124)
00212      WRITE(12,126) GAM,RHO,C,SDIA(KJI),SPERC(KJI),TPR
00213      WRITE(12,125)
00214      WRITE(12,127) RPM,NBLADE,NBSTD,HTR,DTIP,SCHDR(KJI)
00215      WRITE(12,136)
00216      WRITE(12,138) EMR,RSCAL,RVEL,ELT,TIT,AR
00217      WRITE(12,142)
00218      WRITE(12,144) NBLADE,NVANE

```

```

00219      WRITE(12,*)' '
00220      WRITE(12,*)' '
00221      WRITE(12,*)'   CDROTOR = ',CDR
00222      WRITE(12,*)' INLET LENGTH/TIP DIAMETER = ',LINLET
00223  C
00224  C
00225  C   PRELIMINARY CONSTANTS AND COEFFICIENTS
00226  C
00227      DBL=130.+4.342945*ALOG(.105*RHO*(SPCON)*TI**2*A*WWND)
00228      DBLSPL = 4.342945*ALOG(.105*RHO*(SPCON)*TI**2*A*WWND)
00229      WATCON = .105*RHO*(SPCON)*TI**2*A*WWND
00230      PI=3.1415926
00231      TPI=2.*PI
00232      G1OV2=(GAM-1.)/2.
00233      G1OVG=(GAM-1.)/GAM
00234      TDGP1=2./(GAM+1.)
00235      GEXP = (GAM+1.)/(2.*(GAM-1.))
00236      T11=TPR**G1OVG-1.
00237      IF (KJI.EQ.1) TTOT = TABS*(1.+T11/ETAFAN)
00238      T12=1.+1./(G1OV2*EMA**2)
00239      OMM2 =1.-EMA**2
00240      SR1MM2=SQRT(1.-EMA**2)
00241      SR1MR2=SQRT(1.-EMR**2)
00242      CR=EMA/EMRSP
00243      CR2=CR*CR
00244      SR2=1.-CR2
00245      SR=SQRT(SR2)
00246      SRCR=SR*CR
00247      TR=SR/CR
00248      EMRC=EMR/5.
00249  C
00250  C
00251  C   STEADY STATE LIFT COEFFICIENT CALCULATION
00252  C
00253      IF (SCLOPTS .EQ. 1) THEN
00254  C
00255  C CL BASED ON INPUT LIFT COEFFICIENTS
00256  C
00257      CL=SCLS(KJI)
00258      ELSE IF (SCLOPTS .EQ. 2) THEN
00259  C
00260  C CL BASED ON WORK COEFFICIENT
00261  C
00262      CL=2.0*EMT*STHETA(KJI)/(EMR*SIGS)
00263      ELSE IF (SCLOPTS .EQ. 3) THEN
00264  C
00265  C CL BASED ON TOTAL PRESSURE RATIO
00266  C
00267      CL=1.25*(CR2*T12*T11/(SIGS*SR))
00268      ELSE
00269  C
00270  C CL BASED ON STAGGER ANGLE
00271  C
00272      SSAR=SSAD*PI/180.
00273      AINC=SINCDS(KJI)*PI/180.
00274      CLF1=5.65/(1.0+1.8/AR)
00275      CLF2=1.8+AR
00276      CLF3=1.8+AR*SR1MR2
00277      CLPR=CLF1*CLF2/CLF3
00278      CL=CLPR*AINC

```

```

00279          END IF
00280      C
00281      500      AA=1.0-EMA**2
00282              WRITE(12,140) SCLOPTS
00283      C
00284      C      MORE CONSTANTS AND COEFFICIENTS
00285      C
00286              A=0.00
00287              AROT=ATAN(EMT/EMA)
00288              B=CL*SIGS*SR1MR2/(4.*AA)
00289              C=CL*SIGS/4.
00290              FNBL=FLOAT(NBLADE)
00291              FNV =FLOAT(NVANE)
00292              DSNCT=0.5*SIGS*(FNBL/FNV)/(1.-EMA**2)
00293              ALC=TPI*SR1MR2
00294              BETC=TPI
00295              CHIC=BETC*EMT
00296              CHINC = BNEW*(1.-HTR)/(PI*SR1MM2)
00297              DELC=0.00
00298              CDP2=(PI*SIGS/2.0)**2
00299              CAE=1./(EMA*SR1MM2)
00300              NINCO2=6
00301              NINCO2P1=NINCO2+1
00302              NINC=2*NINCO2
00303              FNINC=NINC
00304              DELTH=PI/FNINC
00305              THETA=-PI/2.
00306              IMAX=NINC+1
00307      C
00308      C
00309      C      CALCULATION OF WAVE PROPAGATION FACTORS
00310      C
00311      C
00312              WRITE(12,110)
00313              IF (BW .LE. 0.0) THEN
00314                  WRITE(12,111)
00315              ELSE
00316                  WRITE(12,113) BW
00317              END IF
00318              WRITE(12,112)
00319              WRITE(12,114)
00320      C
00321      C
00322      C      INDEX OVER FREQUENCY - J
00323      C
00324              DO 3000 J=1,NF
00325                  F(J)=FOB(J)/BPF
00326                  ISKIP=0
00327                  IF(BW.GT.0.0.OR.F(J).LT.0.5) ISKIP=1
00328      C
00329              IF (ISKIP .NE. 1) THEN
00330                  FHI=1.122462*F(J)
00331                  FLO=0.890899*F(J)
00332                  FJJ(1)=FLO
00333                  FJJ(2)=F(J)
00334                  FJJ(3)=FHI
00335                  JJMAX=3
00336              ELSE
00337                  FJJ(1)=F(J)
00338                  JJMAX=1

```

```

00339          END IF
00340          CHI=F(J)*CHIC
00341          CHI2=CHI*CHI
00342          NMAXC = CHI*CHINC-0.000001
00343          NMP1C = NMAXC+1
00344          DO IRAD = 1,NMP1C
00345              IRADM1 = IRAD-1
00346              ENR      = FLOAT(IRAD-1)
00347              ENRC     = ENR/CHINC
00348              ENRC2    = ENRC**2
00349              THETA   = -PI/2.
00350              DO I = 1,IMAX
00351                  CTH   = COS(THETA)
00352                  STH   = SIN(THETA)
00353                  CTH2  = CTH**2
00354                  STH2  = STH**2
00355                  IF ((IRAD.GT.1).OR.(I.NE.NINCO2P1)) THEN
00356                      CTF = SQRT (CHI2*STH2+ENRC2*CTH2)
00357                      CTFRAT(IRAD,I) = CHI/CTF
00358                  ELSE
00359                      CTFRAT(IRAD,I) = 1.E+06
00360                  ENDIF
00361                  THETA = THETA+DELTH
00362              ENDDO
00363          ENDDO
00364      C
00365          SDV=0.
00366          SDD=0.
00367          SQV=0.
00368          SQD=0.
00369          DO IJR = 1,NCOF
00370              WSDV(IJR) = 0.00
00371              WSDD(IJR) = 0.00
00372              WSQV(IJR) = 0.00
00373              WSQD(IJR) = 0.00
00374          ENDDO
00375          DO IRAD = 1,NMP1C
00376              DO I = 1,IMAX
00377                  SDVTH(IRAD,I)=0.00
00378                  SDDTH(IRAD,I)=0.00
00379                  SQVTH(IRAD,I)=0.00
00380                  SQDTH(IRAD,I)=0.00
00381              ENDDO
00382          ENDDO
00383          NVAL=30
00384          NVALP1=NVAL+1
00385      C
00386      C
00387      C      INDEX OVER BPF HARMONIC NUMBER    -    N
00388      C
00389          NNCNT      = 0
00390          DO 2800 NN=1,NVALP1
00391              N=NN-1
00392              FLTN=N
00393      1580          EN=N
00394              NNCNT = NNCNT+1
00395              AL=ABS(EN)*ALC
00396              AL2=AL*AL
00397              ALSGNB=AL*SGN(FLTN)*B
00398              BET=EN*BETC/DSTAT

```

```

00399          DEL=DELC*EN
00400          ZQI=0.5*ABS(AL)*SIGS/(1.-EMA**2)
00401          ZQ=ABS(ZQI)
00402          SNQA=SQRT(PI*ZQ*0.5)
00403          SNQDEN=(1.+SNQA)**2
00404      C
00405      C
00406      C          INITIATE BANDWIDTH SUBDIVISION AND NUMERICAL INTEGRATION
00407      C          FOR 1/3-OCTAVE SPECTRUM CALCULATION
00408      C
00409          JJ=1
00410          DO WHILE (JJ .LE. JJMAX)
00411      DO JR = 1,NCOF
00412          WSUMDV(JJ,JR)=0.00
00413          WSUMDD(JJ,JR)=0.00
00414          WSUMQV(JJ,JR)=0.00
00415          WSUMQD(JJ,JR)=0.00
00416      ENDDO
00417      ETARICE = FJJ(JJ)*BPF*DTIP/(12.*SCO(KJI))
00418          CHI=FJJ(JJ)*CHIC
00419          CHI2=CHI*CHI
00420      FVAL=FJJ(JJ)
00421      THE = QDVAL(FVAL,NF,FINT,SPIMPRE,CHECK)
00422      THI = QDVAL(FVAL,NF,FINT,SPIMPR,CHECK)
00423      Z2E = QDVAL(FVAL,NF,FINT,SPIMPZ2E,CHECK)
00424      Z2I = QDVAL(FVAL,NF,FINT,SPIMPZ2,CHECK)
00425      IF (NNCNT.EQ.1) THEN
00426          IF((J.EQ.1).OR.(J.EQ.NF)) THEN
00427              TRME = SQRT(Z2E-THE**2)
00428              TRMI = SQRT(Z2I-THI**2)
00429              TFREQ = FVAL*BPF
00430          ENDIF
00431      ENDIF
00432          SUMDV=0.
00433          SUMDD=0.
00434          SUMQV=0.
00435          SUMQD=0.
00436          THETA=-PI/2.
00437          NMAX = CHI*CHINC-0.000001
00438          NMP1 = NMAX+1
00439          IF ( JJ.EQ.1 ) NMP1L = NMP1
00440      2345          CONTINUE
00441      C          INDEX OVER RADIAL MODE ORDERS
00442          DO 2700 IRAD = 1,NMP1
00443              ENR = FLOAT(IRAD-1)
00444              ENRC = ENR/CHINC
00445              ENRC2 = ENRC**2
00446              NRAD = IRAD-1
00447              CHIN = SQRT ( CHI2-(ENR/CHINC)**2 )
00448              RCHI = CHI/CHIN
00449              KR = (ENR*PI)/(BNEW*(1.-HTR)*CHIN)
00450              SUMDVR=0.
00451              SUMDDR=0.
00452              SUMQVR=0.
00453              SUMQDR=0.
00454          THETA=-PI/2.
00455          DO 1230 I=1,IMAX
00456              CTH=COS(THETA)
00457              EMAMC=EMA*RCHI-CTH
00458              EMAPC=EMA*RCHI+CTH

```

```

00459             IF (I .LE. (NINCO2+1)) THEN
00460                 ZMM(I)=EMAMC
00461                 ZPP(I)=EMAPC
00462                 AEV(I)=CAE/(1.+EMA*CTH)**2
00463                 AED(I)=CAE/(1.-EMA*CTH)**2
00464             END IF
00465 C
00466             STH=SIN(THETA)
00467             TERM=SR1MM2*STH
00468             TERM2=TERM**2
00469             DCV1(I)=TERM2
00470             SNCU(1,I)=ABS(EMAPC)*DSNCT
00471             DCV2(I)=TERM2
00472             SNCU(2,I)=SNCU(1,I)
00473             DCD1(I)=TERM2
00474             SNCD(1,I)=ABS(EMAMC)*DSNCT
00475             DCD2(I)=TERM2
00476             SNCD(2,I)=SNCD(1,I)
00477             STHOSR(I)=STH/SR1MM2
00478             ANG = ATAN(SR1MM2*STH/(CTH+EMA*RCHI))
00479             IF ( ANG.GE.0.00 ) PHI = PI-ANG
00480             IF ( ANG.LT.0.00 ) PHI = -PI-ANG
00481             CALL TRANSROT(PHI,EMA,AROT,TR2,RE2)
00482             TRROT1(I)=TR2
00483             REROT1(I)=RE2
00484             IF ( ANG.GE.0.00 ) PHI = ANG-PI
00485             IF ( ANG.LT.0.00 ) PHI = ANG+PI
00486             CALL TRANSROT(PHI,EMA,AROT,TR2,RE2)
00487             TRROT2(I)=TR2
00488             REROT2(I)=RE2
00489             INDEX=I
00490             IF (I .GT. (NINCO2+1)) INDEX=IMAX+1-I
00491             AEVETC(1,I)=AEV(INDEX)*CDP2*DCV1(I)
00492             AEVETC(2,I)=AEV(INDEX)*CDP2*DCV2(I)
00493             AEDETC(1,I)=AED(INDEX)*CDP2*DCD1(I)
00494             AEDETC(2,I)=AED(INDEX)*CDP2*DCD2(I)
00495             THETA=THETA+DELTH
00496 1230     END DO
00497 C
00498 C
00499 C     INDEX OVER (I) FOR INTEGRATION OVER KY
00500 C
00501             THETA=-PI/2.
00502             DO 2600 I=1,IMAX
00503                 CTH = COS(THETA)
00504                 STH = SIN(THETA)
00505                 CTH2 = CTH**2
00506                 STH2 = STH**2
00507                 IF ((IRAD.GT.1).OR.(I.NE.NINCO2P1)) THEN
00508                     CTF = SQRT (CHI2*STH2+ENRC2*CTH2)
00509                     CTFR = CHI/CTF
00510                 ELSE
00511                     CTFR = 1.E+09
00512                 ENDIF
00513                 CTFR2 = CTFR**2
00514                 IKY = I-(NINCO2+1)
00515                 AKX=-CHI/EMA
00516                 AKY=-BET+CHI*STHOSR(I)
00517                 INDEX=I
00518                 IF (I .GT. (NINCO2+1)) INDEX=IMAX+1-I

```

```

00519             ZM=ZMM( INDEX ) *CHIN
00520             ZP=ZPP( INDEX ) *CHIN
00521             DO 2310 L1=1,2
00522 C
00523 C             INLET TURBULENCE SPECTRUM CALCULATION
00524 C
00525             CALL PHICAL( AKX, AKY, CONTR, EL, RSCAL, RVEL, CR, SR,
00526 &             PHIXX, PHIXY, PHIYY )
00527             TPHIXY=2.*PHIXY
00528 C
00529 C             DIPOLE CONTRIBUTION
00530 C
00531             OMR=SIGS*AKX*DSTAT/2.
00532             OM=ABS( OMR )
00533             CALL FKCAL( OM, AR, SIGS, ELT, EMR, FK )
00534             SRF=FK*FK
00535             SRF2=2.0*FK
00536             OMR=ABS( OMR )
00537             PHIT=PHIXX*SR2-TPHIXY*SRCR+PHIYY*CR2
00538             PHITS=ABS( PHIT*SRF )
00539 C
00540             IF ( L1 .EQ. 2 ) THEN
00541             SNZ  =CHI*SNCU( 2, I )
00542             SNA  =SQRT( PI*SNZ*0.5 )
00543             SNDEN=( 1.+SNA )**2
00544             FNDVM=AEVETC( 2, I ) *PHITS*TRROT2( I ) /SNDEN
00545             FNDVMR=AEVETC( 2, I ) *PHITS*REROT2( I ) /SNDEN
00546             SNZ  =CHI*SNCD( 2, I )
00547             SNA  =SQRT( PI*SNZ*0.5 )
00548             SNDEN=( 1.+SNA )**2
00549             FNDDM=AEDETC( 2, I ) *PHITS/SNDEN+FNDVMR
00550             ELSE
00551             SNZ  =CHI*SNCU( 1, I )
00552             SNA  =SQRT( PI*SNZ*0.5 )
00553             SNDEN=( 1.+SNA )**2
00554             FNDVP=AEVETC( 1, I ) *PHITS*TRROT1( I ) /SNDEN
00555             FNDVPR=AEVETC( 1, I ) *PHITS*REROT1( I ) /SNDEN
00556             SNZ  =CHI*SNCD( 1, I )
00557             SNA  =SQRT( PI*SNZ*0.5 )
00558             SNDEN=( 1.+SNA )**2
00559             FNDDP=AEDETC( 1, I ) *PHITS/SNDEN+FNDVPR
00560             END IF
00561 C
00562 C             QUADRUPOLE CONTRIBUTION
00563 C
00564 2060             ZTERM=ZP
00565             AKXA=AKX*AA
00566             DO 2200 L2=1,2
00567             ZDELA=ZTERM-DEL-AKXA
00568             DEN=( AL2+ZDELA**2 )**2
00569             PART=ZTERM*( ZDELA*A-ALSGNB )+C*ZDELA*
00570 &             CHIN*STHOSR( I ) *AA
00571             GX=ZTERM*PART
00572             GY=CHIN*STHOSR( I ) *PART*AA
00573             TQ=ABS( ( GX**2*PHIXX+GY**2*PHIYY+GX*GY*TPHIXY ) /DEN )
00574             IF ( L2 .EQ. 2 ) THEN
00575             FNQD=TQ*AED( INDEX ) +FNQVR
00576             ELSE
00577             IF( L1.EQ.1 ) TRANS2 = TRROT1( I )
00578             IF( L1.EQ.2 ) TRANS2 = TRROT2( I )

```

```

00579             IF(L1.EQ.1) REFL2 = REROT1(I)
00580             IF(L1.EQ.2) REFL2 = REROT2(I)
00581             FNQV=TQ*AEV(INDEX)*TRANS2
00582             FNQVR=TQ*AEV(INDEX)*REFL2
00583             END IF
00584   C
00585             ZTERM=ZM
00586   2200       END DO
00587   C
00588             IF (L1 .EQ. 2) THEN
00589                 FNQVM=FNQV
00590                 FNQDM=FNQD
00591             ELSE
00592                 FNQVP=FNQV
00593                 FNQDP=FNQD
00594             AKX=CHI/EMA
00595                 ZP=-ZP
00596                 ZM=-ZM
00597             END IF
00598   C
00599   2310       END DO
00600   C
00601             FDV=FNDVP+FNDVM
00602             FDD=FNDDP+FNDDM
00603             FQV=FNQVP+FNQVM
00604             FQD=FNQDP+FNQDM
00605   C
00606             IF ((I .EQ. 1) .OR. (I .EQ. IMAX)) THEN
00607                 FDV=FDV/2.
00608                 FDD=FDD/2.
00609                 FQV=FQV/2.
00610                 FQD=FQD/2.
00611             END IF
00612   C
00613             AKYN = CHIN*STHOSR(I)*BNEW
00614             IF (NNCNT.EQ.1) THEN
00615                 IF(IABSOR.EQ.0)RICEV(JJ,IRAD,I)=1.00
00616                 IF(IABSOR.EQ.0)RICED(JJ,IRAD,I)=1.00
00617             IF(IABSOR.NE.0)THEN
00618                 CTH = COS(THETA)
00619                 STH = SIN(THETA)
00620                 ALP = ( CTH-RCHI*EMA)/OMM2
00621                 ALM = (-CTH-RCHI*EMA)/OMM2
00622                 KYS = STH/SR1MM2
00623                 KTOTP = (RCHI-EMA*ALP)
00624                 KTOTM = (RCHI-EMA*ALM)
00625                 CPHXP = ALP/KTOTP
00626                 CPHXM = ALM/KTOTM
00627                 CPHRP = KR/KTOTP
00628                 CPHRM = KR/KTOTM
00629                 DENP = SQRT(CPHXP**2+CPHRP**2)
00630                 DENM = SQRT(CPHXM**2+CPHRM**2)
00631                 CPHYP = CPHRP/DENP
00632                 CPHYP2 = CPHYP**2
00633                 SPHYP = CPHXP/DENP
00634                 CPHYM = CPHRM/DENM
00635                 CPHYM2 = CPHYM**2
00636                 SPHYM = CPHXM/DENM
00637                 OMSYP = 1.+EMA*SPHYP
00638                 TOMSYP = 2.*OMSYP

```

```

00639          OMSYM = 1.+EMA*SPHYM
00640          TOMSYM = 2.*OMSYM
00641          OMSYP2 = OMSYP**2
00642          OMSYM2 = OMSYM**2
00643          IF ( (I.GT.1).AND.(I.LT.IMAX) ) THEN
00644              TPSIXP = CPHRP/(CPHXP+EMA)
00645              TPSIXM = CPHRM/(CPHXM+EMA)
00646              NBNP = LEXOVD*TPSIXP/(1.-HTR)
00647              NBNM = -LINOVD*TPSIXM/(1.-HTR)
00648              RIXP=(Z2E*CPHY2*OMSYP2-THE*TOMSYP*CPHY+1.)/
00649              &              (Z2E*CPHY2*OMSYP2+THE*TOMSYP*CPHY+1.)
00650              RIXM=(Z2I*CPHYM2*OMSYM2-THI*TOMSYM*CPHYM+1.)/
00651              &              (Z2I*CPHYM2*OMSYM2+THI*TOMSYM*CPHYM+1.)
00652              IXP = RIXP**NBNP
00653              IXM = RIXM**NBNM
00654          ELSE
00655              IXP = 0.00
00656              IXM = 0.00
00657          ENDIF
00658          RICEV(JJ,IRAD,I) = IXM
00659          RICED(JJ,IRAD,I) = IXP
00660          ENDIF
00661          CALL NEWSUB ( RLOW,RUPP,HTR,AKYN,IKY,IRAD,FC )
00662          F3D ( JJ,IRAD,I ) = FC
00663          ENDIF
00664          RICECV = RICEV(JJ,IRAD,I)
00665          RICECD = RICED(JJ,IRAD,I)
00666          F3DF = F3D ( JJ,IRAD,I )
00667          SUMDVR=SUMDVR+FDV*F3DF*RICECV
00668          SUMDDR=SUMDDR+FDD*F3DF*RICECD
00669          SUMQVR=SUMQVR+(FQV/SNQDEN)*F3DF*RICECV
00670          SUMQDR=SUMQDR+(FQD/SNQDEN)*F3DF*RICECD
00671          FDVTH(JJ,IRAD,I) = FDV*F3DF*RICECV
00672          FDDTH(JJ,IRAD,I) = FDD*F3DF*RICECD
00673          FQVTH(JJ,IRAD,I) = FQV*F3DF*RICECV/SNQDEN
00674          FQDTH(JJ,IRAD,I) = FQD*F3DF*RICECD/SNQDEN
00675          TJR1 = (1.-1./CTFR2)
00676          TJR = FCOF*TJR1
00677          JR = 1+INT(TJR)
00678          IF ( JR.GT.NCOF ) JR = NCOF
00679          WSUMDV(JJ,JR)=WSUMDV(JJ,JR)+FDVTH(JJ,IRAD,I)
00680          WSUMDD(JJ,JR)=WSUMDD(JJ,JR)+FDDTH(JJ,IRAD,I)
00681          WSUMQV(JJ,JR)=WSUMQV(JJ,JR)+FQVTH(JJ,IRAD,I)
00682          WSUMQD(JJ,JR)=WSUMQD(JJ,JR)+FQDTH(JJ,IRAD,I)
00683          THETA=THETA+DELTH
00684          2600          END DO
00685          SUMDV=SUMDV+SUMDVR
00686          SUMDD=SUMDD+SUMDDR
00687          SUMQV=SUMQV+SUMQVR
00688          SUMQD=SUMQD+SUMQDR
00689          2700          CONTINUE
00690          C
00691          PROD = CHI2*DELTH
00692          SNDV=PROD*SUMDV
00693          SNDD=PROD*SUMDD
00694          SNQV=SUMQV*DELTH
00695          SNQD=SUMQD*DELTH
00696          DO IJR = 1,NCOF
00697              WSNDV(JJ,IJR)=PROD*WSUMDV(JJ,IJR)
00698              WSNDD(JJ,IJR)=PROD*WSUMDD(JJ,IJR)

```

```

00699          WSNQV(JJ,IJR)=DELTH*WSUMQV(JJ,IJR)
00700          WSNQD(JJ,IJR)=DELTH*WSUMQD(JJ,IJR)
00701      ENDDO
00702  C
00703      DO IRAD = 1,NMP1
00704          DO I = 1,IMAX
00705              FDVTH(JJ,IRAD,I) = PROD*FDVTH(JJ,IRAD,I)
00706              FDDTH(JJ,IRAD,I) = PROD*FDDTH(JJ,IRAD,I)
00707              FQVTH(JJ,IRAD,I) = FQVTH(JJ,IRAD,I)*DELTH
00708              FQDTH(JJ,IRAD,I) = FQDTH(JJ,IRAD,I)*DELTH
00709          ENDDO
00710      ENDDO
00711  C
00712      DVP(JJ)=SNDV
00713      DDP(JJ)=SNDD
00714      QVP(JJ)=SNQV
00715      QDP(JJ)=SNQD
00716      DO IRAD = 1,NMP1
00717          DO I = 1,IMAX
00718              NDVTH(IRAD,I)=FDVTH(JJ,IRAD,I)
00719              NDDTH(IRAD,I)=FDDTH(JJ,IRAD,I)
00720              NQVTH(IRAD,I)=FQVTH(JJ,IRAD,I)
00721              NQDTH(IRAD,I)=FQDTH(JJ,IRAD,I)
00722          ENDDO
00723      ENDDO
00724      DO IJR = 1,NCOF
00725          WSNDV1(IJR) = WSNDV(JJ,IJR)
00726          WSNDD1(IJR) = WSNDJ(JJ,IJR)
00727          WSNQV1(IJR) = WSNQV(JJ,IJR)
00728          WSNQD1(IJR) = WSNQD(JJ,IJR)
00729      ENDDO
00730  C
00731          IF (ISKIP .NE. 1) THEN
00732              DVP(JJ)=DVP(JJ)/FJJ(JJ)
00733              DDP(JJ)=DDP(JJ)/FJJ(JJ)
00734              QVP(JJ)=QVP(JJ)/FJJ(JJ)
00735              QDP(JJ)=QDP(JJ)/FJJ(JJ)
00736          DO IJR = 1,NCOF
00737              WSNDV(JJ,IJR) = WSNDV(JJ,IJR)/FJJ(JJ)
00738              WSNDD(JJ,IJR) = WSNDJ(JJ,IJR)/FJJ(JJ)
00739              WSNQV(JJ,IJR) = WSNQV(JJ,IJR)/FJJ(JJ)
00740              WSNQD(JJ,IJR) = WSNQD(JJ,IJR)/FJJ(JJ)
00741          ENDDO
00742          END IF
00743          IF (ISKIP .NE. 1) THEN
00744              DO IRAD = 1,NMP1
00745                  DO I = 1,IMAX
00746                      FDVTH(JJ,IRAD,I) = FDVTH(JJ,IRAD,I)/FJJ(JJ)
00747                      FDDTH(JJ,IRAD,I) = FDDTH(JJ,IRAD,I)/FJJ(JJ)
00748                      FQVTH(JJ,IRAD,I) = FQVTH(JJ,IRAD,I)/FJJ(JJ)
00749                      FQDTH(JJ,IRAD,I) = FQDTH(JJ,IRAD,I)/FJJ(JJ)
00750                  ENDDO
00751              ENDDO
00752          END IF
00753  C
00754          JJ=JJ+1
00755      END DO
00756  C
00757      IF (ISKIP .NE. 1) THEN
00758          IF (JJMAX .EQ. 3) THEN

```

```

00759      FJ=F(J)
00760      CALL EXINT(DVP(1),DVP(2),FLO,FJ,SNDRV1)
00761      CALL EXINT(DVP(2),DVP(3),FJ,FHI,SNDRV2)
00762          SNDRV=SNDRV1+SNDRV2
00763      DO IJR = 1,NCOF
00764          W1 = WSNDV(1,IJR)
00765          W2 = WSNDV(2,IJR)
00766          W3 = WSNDV(3,IJR)
00767          CALL EXINT(W1,W2,FLO,FJ,R1)
00768          CALL EXINT(W2,W3,FJ,FHI,R2)
00769          WSNDV1(IJR) = R1+R2
00770      ENDDO
00771      CALL EXINT(DDP(1),DDP(2),FLO,FJ,SNDD1)
00772      CALL EXINT(DDP(2),DDP(3),FJ,FHI,SNDD2)
00773          SNDD=SNDD1+SNDD2
00774      DO IJR = 1,NCOF
00775          W1 = WSNDD(1,IJR)
00776          W2 = WSNDD(2,IJR)
00777          W3 = WSNDD(3,IJR)
00778          CALL EXINT(W1,W2,FLO,FJ,R1)
00779          CALL EXINT(W2,W3,FJ,FHI,R2)
00780          WSNDD1(IJR) = R1+R2
00781      ENDDO
00782      CALL EXINT(QVP(1),QVP(2),FLO,FJ,SNQV1)
00783      CALL EXINT(QVP(2),QVP(3),FJ,FHI,SNQV2)
00784          SNQV=SNQV1+SNQV2
00785      DO IJR = 1,NCOF
00786          W1 = WSNQV(1,IJR)
00787          W2 = WSNQV(2,IJR)
00788          W3 = WSNQV(3,IJR)
00789          CALL EXINT(W1,W2,FLO,FJ,R1)
00790          CALL EXINT(W2,W3,FJ,FHI,R2)
00791          WSNQV1(IJR) = R1+R2
00792      ENDDO
00793      CALL EXINT(QDP(1),QDP(2),FLO,FJ,SNQD1)
00794      CALL EXINT(QDP(2),QDP(3),FJ,FHI,SNQD2)
00795          SNQD=SNQD1+SNQD2
00796      DO IJR = 1,NCOF
00797          W1 = WSNQD(1,IJR)
00798          W2 = WSNQD(2,IJR)
00799          W3 = WSNQD(3,IJR)
00800          CALL EXINT(W1,W2,FLO,FJ,R1)
00801          CALL EXINT(W2,W3,FJ,FHI,R2)
00802          WSNQD1(IJR) = R1+R2
00803      ENDDO
00804      ELSE
00805          CALL EXINT(DVP(1),DVP(2),FLO,FHI,SNDRV)
00806      DO IJR = 1,NCOF
00807          W1 = WSNDV(1,IJR)
00808          W2 = WSNDV(2,IJR)
00809          CALL EXINT(W1,W2,FLO,FHI,R1)
00810          WSNDV1(IJR) = R1
00811      ENDDO
00812          CALL EXINT(DDP(1),DDP(2),FLO,FHI,SNDD)
00813      DO IJR = 1,NCOF
00814          W1 = WSNDD(1,IJR)
00815          W2 = WSNDD(2,IJR)
00816          CALL EXINT(W1,W2,FLO,FHI,R1)
00817          WSNDD1(IJR) = R1
00818      ENDDO

```

```

00819          CALL EXINT(QVP(1),QVP(2),FLO,FHI,SNQV)
00820      DO IJR = 1,NCOF
00821          W1 = WSNQV(1,IJR)
00822          W2 = WSNQV(2,IJR)
00823          CALL EXINT(W1,W2,FLO,FHI,R1)
00824          WSNQV1(IJR) = R1
00825      ENDDO
00826          CALL EXINT(QDP(1),QDP(2),FLO,FHI,SNQD)
00827      DO IJR = 1,NCOF
00828          W1 = WSNQD(1,IJR)
00829          W2 = WSNQD(2,IJR)
00830          CALL EXINT(W1,W2,FLO,FHI,R1)
00831          WSNQD1(IJR) = R1
00832      ENDDO
00833          END IF
00834      C
00835          END IF
00836      C
00837      IF (ISKIP .NE. 1) THEN
00838          IF (JJMAX .EQ. 3) THEN
00839      DO IRAD = 1,NMP1
00840          DO I = 1,IMAX
00841              DVP(1) = FDVTH(1,IRAD,I)
00842              DDP(1) = FDDTH(1,IRAD,I)
00843              QVP(1) = FQVTH(1,IRAD,I)
00844              QDP(1) = FQDTH(1,IRAD,I)
00845              DVP(2) = FDVTH(2,IRAD,I)
00846              DDP(2) = FDDTH(2,IRAD,I)
00847              QVP(2) = FQVTH(2,IRAD,I)
00848              QDP(2) = FQDTH(2,IRAD,I)
00849              DVP(3) = FDVTH(3,IRAD,I)
00850              DDP(3) = FDDTH(3,IRAD,I)
00851              QVP(3) = FQVTH(3,IRAD,I)
00852              QDP(3) = FQDTH(3,IRAD,I)
00853              CALL EXINT(DVP(1),DVP(2),FLO,FJ,SNQV1)
00854              CALL EXINT(DVP(2),DVP(3),FJ,FHI,SNQV2)
00855              NDVTH(IRAD,I) = SNQV1+SNQV2
00856              CALL EXINT(DDP(1),DDP(2),FLO,FJ,SNDD1)
00857              CALL EXINT(DDP(2),DDP(3),FJ,FHI,SNDD2)
00858              NDDTH(IRAD,I) = SNDD1+SNDD2
00859              CALL EXINT(QVP(1),QVP(2),FLO,FJ,SNQV1)
00860              CALL EXINT(QVP(2),QVP(3),FJ,FHI,SNQV2)
00861              NQVTH(IRAD,I) = SNQV1+SNQV2
00862              CALL EXINT(QDP(1),QDP(2),FLO,FJ,SNQD1)
00863              CALL EXINT(QDP(2),QDP(3),FJ,FHI,SNQD2)
00864              NQDTH(IRAD,I) = SNQD1+SNQD2
00865          ENDDO
00866          ENDDO
00867      ELSE
00868      DO IRAD = 1,NMP1
00869          DO I = 1,IMAX
00870              DVP(1) = FDVTH(1,IRAD,I)
00871              DDP(1) = FDDTH(1,IRAD,I)
00872              QVP(1) = FQVTH(1,IRAD,I)
00873              QDP(1) = FQDTH(1,IRAD,I)
00874              DVP(2) = FDVTH(2,IRAD,I)
00875              DDP(2) = FDDTH(2,IRAD,I)
00876              QVP(2) = FQVTH(2,IRAD,I)
00877              QDP(2) = FQDTH(2,IRAD,I)
00878              CALL EXINT(DVP(1),DVP(2),FLO,FHI,DUM)

```

```

00879          NDVTH(IRAD,I)      = DUM
00880          CALL EXINT(DDP(1),DDP(2),FLO,FHI,DUM)
00881          NDDTH(IRAD,I)      = DUM
00882          CALL EXINT(QVP(1),QVP(2),FLO,FHI,DUM)
00883          NQVTH(IRAD,I)      = DUM
00884          CALL EXINT(QDP(1),QDP(2),FLO,FHI,DUM)
00885          NQDTH(IRAD,I)      = DUM
00886          ENDDO
00887          ENDDO
00888          END IF
00889      C
00890          END IF
00891      C
00892          IF (N .GT. 0) THEN
00893              SDV=SDV+4.*SNDV
00894              SDD=SDD+4.*SNDD
00895              SQV=SQV+4.*SNQV
00896              SQD=SQD+4.*SNQD
00897          DO IJR = 1,NCOF
00898              WSDV(IJR)=WSDV(IJR)+4.*WSNDV1(IJR)
00899              WSDD(IJR)=WSDD(IJR)+4.*WSNDD1(IJR)
00900              WSQV(IJR)=WSQV(IJR)+4.*WSNQV1(IJR)
00901              WSQD(IJR)=WSQD(IJR)+4.*WSNQD1(IJR)
00902          ENDDO
00903          ELSE
00904              SDV=SDV+SNDV
00905              SDD=SDD+SNDD
00906              SQV=SQV+SNQV
00907              SQD=SQD+SNQD
00908          DO IJR = 1,NCOF
00909              WSDV(IJR)=WSDV(IJR)+WSNDV1(IJR)
00910              WSDD(IJR)=WSDD(IJR)+WSNDD1(IJR)
00911              WSQV(IJR)=WSQV(IJR)+WSNQV1(IJR)
00912              WSQD(IJR)=WSQD(IJR)+WSNQD1(IJR)
00913          ENDDO
00914          END IF
00915          IF (N .GT. 0) THEN
00916          DO IRAD = 1,NMP1
00917              DO I = 1,IMAX
00918                  SDVTH(IRAD,I)=SDVTH(IRAD,I)+4.*NDVTH(IRAD,I)
00919                  SDDTH(IRAD,I)=SDDTH(IRAD,I)+4.*NDDTH(IRAD,I)
00920                  SQVTH(IRAD,I)=SQVTH(IRAD,I)+4.*NQVTH(IRAD,I)
00921                  SQDTH(IRAD,I)=SQDTH(IRAD,I)+4.*NQDTH(IRAD,I)
00922              ENDDO
00923          ENDDO
00924          ELSE
00925          DO IRAD = 1,NMP1
00926              DO I = 1,IMAX
00927                  SDVTH(IRAD,I)=SDVTH(IRAD,I)+NDVTH(IRAD,I)
00928                  SDDTH(IRAD,I)=SDDTH(IRAD,I)+NDDTH(IRAD,I)
00929                  SQVTH(IRAD,I)=SQVTH(IRAD,I)+NQVTH(IRAD,I)
00930                  SQDTH(IRAD,I)=SQDTH(IRAD,I)+NQDTH(IRAD,I)
00931              ENDDO
00932          ENDDO
00933          END IF
00934      C
00935      2800      END DO
00936      C
00937          SDV=SDV*EMRC
00938          SDD=SDD*EMRC

```

```

00939          SQV=SQV*EMRC
00940          SQD=SQD*EMRC
00941          STV=SDV+SQV
00942          STD=SDD+SQD
00943      DO IJR = 1,NCOF
00944          WSDV(IJR)=WSDV(IJR)*EMRC
00945          WSDD(IJR)=WSDD(IJR)*EMRC
00946          WSQV(IJR)=WSQV(IJR)*EMRC
00947          WSQD(IJR)=WSQD(IJR)*EMRC
00948          WSTV(IJR)=WSDV(IJR)+WSQV(IJR)
00949          WSTD(IJR)=WSDD(IJR)+WSQD(IJR)
00950      ENDDO
00951      TRMDV1= 0.00
00952      TRMDD1= 0.00
00953      TRMQV1= 0.00
00954      TRMQD1= 0.00
00955      TRMV1 = 0.00
00956      TRMD1 = 0.00
00957      DO I = 1,NCOF
00958          TRMDV1= WSDV(I)+TRMDV1
00959          TRMDD1= WSDD(I)+TRMDD1
00960          TRMQV1= WSQV(I)+TRMQV1
00961          TRMQD1= WSQD(I)+TRMQD1
00962          TRMV1 = WSTV(I)+TRMV1
00963          TRMD1 = WSTD(I)+TRMD1
00964      ENDDO
00965      DO IRAD = 1,NMP1
00966          DO I = 1,IMAX
00967              SDVTH(IRAD,I)=SDVTH(IRAD,I)*EMRC
00968              SDDTH(IRAD,I)=SDDTH(IRAD,I)*EMRC
00969              SQVTH(IRAD,I)=SQVTH(IRAD,I)*EMRC
00970              SQDTH(IRAD,I)=SQDTH(IRAD,I)*EMRC
00971              STVTH(IRAD,I)=SDVTH(IRAD,I)+SQVTH(IRAD,I)
00972              STDTH(IRAD,I)=SDDTH(IRAD,I)+SQDTH(IRAD,I)
00973          ENDDO
00974      ENDDO
00975          FHZ=FOB(J)
00976          DBNB=0.0
00977      IF (J.EQ.NTOBNI) THEN
00978          WRITE(13,*) ' '
00979          POWMAX = STVTH(1,1)+SDVTH(1,1)
00980          ICOUNT = 0
00981          DO IRAD = 1,NMP1
00982              DO I = 1,IMAX
00983                  ICOUNT = ICOUNT+1
00984                  CTFRATN ( ICOUNT ) = CTFRAT ( IRAD,I )
00985                  POWTOT ( ICOUNT ) = STVTH ( IRAD,I ) +
00986                      & STDTH ( IRAD,I )
00987                  IF(POWTOT(ICOUNT).GT.POWMAX)
00988                      & POWMAX = POWTOT( ICOUNT )
00989              ENDDO
00990          ENDDO
00991          DO ISP = 1,ICOUNT
00992              IPERM ( ISP ) = ISP
00993          ENDDO
00994          CALL SVRGP ( ICOUNT,CTFRATN,CTFRATO,IPERM )
00995          DO IRAD = 1,NMP1
00996              DO I = 1,IMAX
00997                  CTFINT = CTFRAT ( IRAD,I )
00998                  POWINT = STVTH ( IRAD,I ) + STDTH ( IRAD,I )

```

```

00999          IF (POWINT.GT.0.00) THEN
01000              POWDB  = 10.00*ALOG10(POWINT/POWMAX)
01001          ELSE
01002              POWDB  = -1.00E+06
01003          ENDIF
01004      ENDDO
01005  ENDDO
01006  DO ISP = 1,ICOUNT
01007      CTFINI = CTFRATO ( ISP )
01008      IREL   = IPERM ( ISP )
01009      POWRAT = POWTOT ( IREL ) / POWMAX
01010      IF (POWRAT.GT.0.00) THEN
01011          POWDB  = 10.00*ALOG10(POWRAT)
01012      ELSE
01013          POWDB  = -1.00E+06
01014      ENDIF
01015  ENDDO
01016  SUMCHKDV = 0.00
01017  SUMCHKDD = 0.00
01018  SUMCHKQV = 0.00
01019  SUMCHKQD = 0.00
01020  SUMCHKV  = 0.00
01021  SUMCHKD  = 0.00
01022  DO IRAD = 1,NMP1
01023      DO I = 1,IMAX
01024          SUMCHKDV = SUMCHKDV+SDVTH( IRAD,I)
01025          SUMCHKDD = SUMCHKDD+SDDTH( IRAD,I)
01026          SUMCHKQV = SUMCHKQV+SQVTH( IRAD,I)
01027          SUMCHKQD = SUMCHKQD+SQDTH( IRAD,I)
01028          SUMCHKV  = SUMCHKV+STVTH( IRAD,I)
01029          SUMCHKD  = SUMCHKD+STDTH( IRAD,I)
01030      ENDDO
01031  ENDDO
01032      WRITE(13,*) '              SUMCHK      ',
01033      &          '          SUM'
01034      WRITE(13,*) ' '
01035      WRITE(13,*) 'UPSTR DIPOLE CHECKS ',SUMCHKDV,SDV,TRMDV1
01036      WRITE(13,*) 'DNSTR DIPOLE CHECKS ',SUMCHKDD,SDD,TRMDD1
01037      WRITE(13,*) 'UPSTR QUADRU CHECKS ',SUMCHKQV,SQV,TRMQV1
01038      WRITE(13,*) 'DNSTR QUADRU CHECKS ',SUMCHKQD,SQD,TRMQD1
01039      WRITE(13,*) 'UPSTR TOTAL  CHECKS ',SUMCHKV,STV,TRMV1
01040      WRITE(13,*) 'DNSTR TOTAL  CHECKS ',SUMCHKD,STD,TRMD1
01041  ENDIF
01042  C
01043      IF (BW .LE. 0.0) THEN
01044          IF (F(J) .LT. 0.5) DBNB= -6.35
01045      ELSE
01046          DBNB=10.0*ALOG10(BW/FOB(J))
01047      END IF
01048  DBNBT = 10.** (0.1*DBNB)
01049  C
01050      IF ( SDV.GT.0.00 ) THEN
01051          SDVDB=10.0*ALOG10(SDV) + DBL + DBNB
01052      ELSE
01053          SDVDB=-100.
01054      ENDIF
01055      SDDDB=10.0*ALOG10(SDD) + DBL + DBNB
01056      IF ( SQV.GT.0.00 ) THEN
01057          SQVDB=10.0*ALOG10(SQV) + DBL + DBNB
01058      ELSE

```

```

01059          SQVDB=-100.
01060      ENDIF
01061          SQDDB=10.0*ALOG10(SQD) + DBL + DBNB
01062      IF ( STV.GT.0.00 ) THEN
01063          STVDB=10.0*ALOG10(STV) + DBL + DBNB
01064      ELSE
01065          STVDB=-100.
01066      ENDIF
01067          STDDB=10.0*ALOG10(STD) + DBL + DBNB
01068      IF ( J.EQ.NTOBNI ) WRITE(13,* ) ' OBN = ',TOBN(J)
01069      IF ( J.EQ.NTOBNI ) WRITE(13,* ) ' '
01070      DO IJR = 1,NCOF
01071          WUP(IJR,J) = WUP(IJR,J)+ DBNBT*WSTV(IJR)*WATCON
01072          WDN(IJR,J) = WDN(IJR,J)+ DBNBT*WSTD(IJR)*WATCON
01073      ENDDO
01074          PVT(J)=10.0*ALOG10(10.0**(PVT(J)/10.0)+
01075      &          10.0**(STVDB/10.0))
01076          PDT(J)=10.0*ALOG10(10.0**(PDT(J)/10.0)+
01077      &          10.0**(STDDB/10.0))
01078      C
01079      C
01080          WRITE(12,116) FHZ,F(J),SDVDB,SDDDB,SQVDB,
01081      &          SQDDB,STVDB,STDDB
01082      C
01083      3000      END DO
01084      C
01085      1949      END DO
01086      C
01087      C
01088      C      FAN TOTAL POWER SPECTRUM
01089      C
01090          WRITE(12,132)
01091          TPR      = STPRIN(MIDSTR)
01092          TABS      = (SCO(MIDSTR)/49.0422)**2
01093          PABS      = 53.3*TABS*RHO/144.
01094          PTOT      = PABS*TPR
01095          T11      = TPR**G1OVG-1.
01096          TTOT      = TABS*(1.+T11/ETAFAN)
01097          FMACH      = -ABS(SEMA(MIDSTR))
01098          FMACHS      = -ABS(MACHS)
01099          FMACHD      = ABS(SEMA(MIDSTR))
01100          XM          = FMACHD
01101          XM2          = XM**2
01102          AOAST      = (TDGP1*(1.+G1OV2*XM2))**GEXP/XM
01103          ANOZRATC      = 1./AOAST
01104          IF ( ANOZRATC.GT.ANOZRAT ) ANOZRAT = 1.02*ANOZRATC
01105          FMACH2      = ABS(MACHS)
01106          DO 101 J=1,NF
01107              ETARICE = FOB(J)*DTIP/(12.*SCO(MIDSTR))
01108              PVDBT=PVT(J)
01109              PDDBT=PDT(J)
01110              PTDB=10.0*ALOG10(10.0**(PVDBT/10.0)+
01111      &          10.0**(PDDBT/10.0))
01112              WRITE(12,134) NINT(TOBN(J)), F(J) ,PVDBT,PDDBT,PTDB
01113              DO IJR = 1,NCOF
01114                  WSUMIN(IJR) = WUP (IJR,J)
01115              END DO
01116              CALL BBRDCFIN(TABS,PABS,RADMIC,ISIDELN,DTIP,ALIP,BLIP,FMACH,
01117      &FMACHS,NCOF,WSUMIN,ETARICE,DELANG,NANGLE,ANGLE,
01118      &SPLOUT,SPLTL,WATTS,WATTRAN)

```

```

01119      WATTDB = 130.00 + 10.00*ALOG10(WATTS)
01120      WRITE(13,*)' INPOWER CHECK IN DB',WATTDB,TOBN(J),PVDBT
01121      NANGI   = NANGLE
01122      DO IANG = 1,NANGI
01123          SPLVDB(IANG) = SPLOUT(IANG)
01124      ENDDO
01125      DO IJR = 1,NCOF
01126          WSUMIN(IJR) = WDN (IJR,J)
01127      END DO
01128      CALL BBRDCFEX(TTOT,PTOT,TABS,PABS,HTR,ANOZRAT,RADMIC,
01129      & ISIDELN,DJIP,DJET,FMACHD,FMACH1,FMACH2,NCOF,WSUMIN,DELANG,
01130      & ETARICE,NANGLE,ANGLE,SPLOUT,SPLTL,WATTS,WATTRAN,FMACHN,
01131      & COFMIN)
01132      WATTDB = 130.00 + 10.00*ALOG10(WATTS)
01133      WRITE(13,*)' EXPOWER CHECK IN DB',WATTDB,TOBN(J),PDDBT
01134      NANG   = NANGLE
01135      DO IANG = 1,NANG
01136          II           = NANGLE+1-IANG
01137          SPLDDB(IANG) = SPLOUT(II)
01138          ANGLEO(IANG) = ANGLE(II)
01139          IF ( IANG.GT.NANGI ) SPLVDB ( IANG ) = -150.00
01140          P2IN          = 10.**(0.1*SPLVDB(IANG))
01141          P2EX          = 10.**(0.1*SPLDDB(IANG))
01142          SPLDBT(IANG) = 10.*ALOG10(P2IN+P2EX)
01143      ENDDO
01144      IF ( J.EQ.NTOBNI ) THEN
01145          DO IANG = 1,NANG
01146              IF ( IANG.EQ.1 ) WRITE(13,*)' TOTAL OVER ALL STRIPS'
01147              IF ( IANG.EQ.1 ) WRITE(13,*)
01148              IF ( IANG.EQ.1 ) WRITE(13,150)
01149              IF ( IANG.EQ.1 ) WRITE(13,*)
01150              WRITE(13,160)ANGLEO(IANG),SPLVDB(IANG),
01151      &          SPLDDB(IANG),SPLDBT(IANG)
01152          ENDDO
01153      ENDIF
01154      C
01155      C      .. Write data to spl plot file
01156      C
01157          FREQNCY = 10.00**(0.1*TOBN(J))
01158          DO IANG = 1,NANG
01159              IF ( IANG.EQ.1 ) WRITE(14,148)INT(FREQNCY),INT(TOBN(J))
01160              IF ( IANG.EQ.1 ) WRITE(14,*)
01161              IF ( IANG.EQ.1 ) WRITE(14,150)
01162              IF ( IANG.EQ.1 ) WRITE(14,*)
01163              WRITE(14,160)ANGLEO(IANG),SPLVDB(IANG),
01164      &          SPLDDB(IANG),SPLDBT(IANG)
01165          ENDDO
01166      C
01167      101      END DO
01168      C
01169      END DO
01170      GO TO 9999
01171      C
01172      C****      ERROR DURING READ
01173      C
01174      1000 WRITE(12,1002)
01175      C
01176      C      FORMAT SECTION
01177      C
01178      106      FORMAT(/,27H                      CASE NUMBER,I4,5H      OF,I4)

```

```

01179 108  FORMAT(/,16X,22HPROGRAM *** STATIN ***/
01180      &13X,39HRESPONSE OF AN ISOLATED OGV LIKE STATOR
01181      &/,12X,32HTO INGESTION OF INLET TURBULENCE)
110  FORMAT(/, ' *** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13)',
& ' WATS ***')
01184 111  FORMAT(/, ' ONE THIRD OCTAVE')
01185 112  FORMAT(/, ' FREQUENCY DIPOLE QUADRUPOLE',
01186      & ' TOTAL')
01187 113  FORMAT(/, ' BANDWIDTH = ',F6.1,' Hz')
114  FORMAT(' HERTZ F/BPF INLET EXHAUST INLET EXHAUST INLET',
& ' EXHAUST')
01190 116  FORMAT(I8,F8.4,F8.1)
01191 118  FORMAT(/,24H***** STRIP AREA NUMBER ,I4,9H WWND = ,F6.4)
120  FORMAT(/,52H EMA EMTIP TI SINCD CONTR L/SSTD)
122  FORMAT(2F9.3,F9.4,F9.2,F8.3,F7.2)
124  FORMAT(/,53H GAM RHO C SDIA SPERC TPR)
01195 125  FORMAT(/,53H RPM NB NBSTD HTR DTIP CHDR)
126  FORMAT(F9.3,F9.4,F9.0,F9.3,F8.3,F9.3)
01197 127  FORMAT(F9.1,I9,I9,F9.3,F8.3,F9.3)
01198 128  FORMAT(/,3X,6H CLW=F08.4,4X,6HCLINC=F08.4,4X,6HCLINP=F08.4)
01199 132  FORMAT(/,12X,24HFAN TOTAL POWER SPECTRUM//
01200      &7X,4HTOBN,8X,5HF/BPF,6X,6HPWL-UP,6X,6HPWL-DN,5X,7HPWL-TOT,4X)
01201 134  FORMAT(I11,F13.4,3F12.2)
136  FORMAT(/,53H EMR RSCAL RVEL ELT TIT AR )
138  FORMAT(F9.3,F9.2,F9.4,F9.3,F9.4,F9.3)
01204 140  FORMAT(/, ' STREAMLINE LIFT COEFFICIENT CALCULATED U
01205      &SING SCLOPT= ',I2)
01206 142  FORMAT(/,30X,2HNB,4X,2HNS)
01207 144  FORMAT(29X,I3,3X,I3)
01208 148  FORMAT(/,1X,'FREQUENCY =',I6,', OBN =',I3)
01209 150  FORMAT(1X,'ANGLE',1X,'INL SPL',1X,'EXH SPL',1X,'TOT SPL')
01210 160  FORMAT(1X,F5.1,1X,3(1X,F7.1))
01211 1002 FORMAT(/,6X,41H** INPUT ERROR ** PROCEEDING TO NEXT CASE//)
01212 C
01213 9999 CLOSE (UNIT=11)
01214 CLOSE (UNIT=12)
01215 CLOSE (UNIT=13)
01216 CLOSE (UNIT=14)
01217 10001 CONTINUE
01218 C
01219 STOP
01220 END

00001 C
00002 C FKCAL CALCULATION OF FK
00003 C
00004 SUBROUTINE FKCAL(OM,AR,SIGS,ELT,EMR,FK)
00005 C
00006 PI=3.1415926
00007 C
00008 IF(EMR.GT.0.8) GOTO 15
00009 BETASQ=1.0-EMR*EMR
00010 OMS=OM/BETASQ
00011 AMU=EMR*OMS
00012 IF(AMU.GT.1.0) GOTO 15
00013 BETA=SQRT(BETASQ)
00014 OMK=OMS*EMR*EMR
00015 SEARS=SQRT(1.0/(2.0*PI*OMS+1.0/(1.0+2.4*OMS)))
00016 SEARS=SEARS/BETA
00017 SEARS=SEARS*SQRT(1.0-(0.5*OMK)**2)

```

```

00018      GOTO 25
00019      15 CONTINUE
00020      EX=2.0*EMR*OM/(1.0+EMR)
00021      Z =SQRT(2.0*EX/PI)
00022      CALL FRESNL(Z,C2X,S2X)
00023      SEARS=1.0/(PI*OM)
00024      SEARS=SEARS*SQRT(2.0*(C2X**2+S2X**2)/EMR)
00025      25 CONTINUE
00026      ELTOH=ELT/(AR*SIGS)
00027      A=1.0/(2.0*ELTOH)
00028      ASQ=A*A
00029      BSQ=(AR**2)*(OM**2+2.0/PI**2)
00030      B=SQRT(BSQ)
00031      DEN=BSQ-ASQ
00032      ANUM=0.0
00033      C
00034      IF (A .LE. 25.0) ANUM=EXP(-2.0*A) -1.0
00035      ACON=ANUM/DEN
00036      BNUM=0.0
00037      C
00038      IF (B .LE. 25.0) BNUM=EXP(-2.0*B) -1.0
00039      BCON=BNUM/DEN
00040      FKSQ=1.0/A + 0.5*(BSQ/ASQ)*ACON - 0.5*(A/B)*BCON
00041      AFKSQ=ABS(FKSQ)
00042      FK=SEARS*SQRT(AFKSQ)
00043      C
00044      RETURN
00045      END

00001      C
00002      C          PHICAL          CALCULATION OF PHIXX, PHIXY, AND PHIYY
00003      C
00004      SUBROUTINE PHICAL(AKXI,AKYI,CONTR,EL,RSCAL,RVEL,CR,SR,
00005      &                PHIXXO,PHIXYO,PHIYYO)
00006      C
00007      PI=3.1415926
00008      RVEL2=RVEL**2
00009      RSCAL2=RSCAL**2
00010      C=CONTR
00011      IF(C.LE.0.0) C=1.0
00012      EPS=1.0/C**3
00013      OMEPS=1.0-EPS
00014      OMEPS2=OMEPS**2
00015      CR2=CR**2
00016      SR2=SR**2
00017      TCRSR=2.*CR*SR
00018      AKX=AKXI*CR+AKYI*SR
00019      AKY=-AKXI*SR+AKYI*CR
00020      C
00021      ELA=EL
00022      ELT=EL/RSCAL
00023      AK1=AKX*C
00024      AK2=AKY/SQRT(C)
00025      AKNX=AK1*ELA
00026      AKNY=AK2*ELT
00027      AKNX2=AKNX**2
00028      AKNY2=AKNY**2
00029      C
00030      ALT2=1.0+AKNX2+AKNY2
00031      ALT=SQRT(ALT2)

```

```

00032      RA5=1.0/ALT**5
00033      A=ALT/ELT
00034      C
00035      FPHIYY=2.0*RVEL2-1.0/RSCAL2
00036      IF (C .LE. 1.01 .AND. C .GE. 0.99) THEN
00037      C
00038      C      ANALYTIC TWO-DIMENSIONAL SPECTRA FOR CONTRACTION RATIO = 1.0
00039      C
00040      CPHIXX=ELT*ELA*RA5/(4.0*PI)
00041      CPHIXY=-3.0*CPHIXX/RSCAL
00042      CPHIYY=CPHIXX
00043      C
00044      PHIXX=CPHIXX*(3.0*AKNY2+ALT2)
00045      PHIXY=CPHIXY*(AKNX*AKNY)
00046      PHIYY=CPHIYY*(3.0*AKNX2/RSCAL2+ALT2*FPHIYY)
00047      PHIXXO=PHIXX*CR2-TCRSR*PHIXY+PHIYY*SR2
00048      PHIXYO=(PHIXX-PHIYY)*CR*SR+PHIXY*(CR2-SR2)
00049      PHIYYO=PHIXX*SR2+TCRSR*PHIXY+PHIYY*CR2
00050      RETURN
00051      END IF
00052      C
00053      C      NUMERICAL INTEGRATION OVER K3 FOR CONTRACTION RATIO .NE. 1.0
00054      C
00055      RC2 =1.0/C**2
00056      RSQC=1.0/SQRT(C)
00057      CPHI=4.0*SQRT(C)*ELA*ELT/PI**2
00058      PHI11=0.0
00059      PHI12=0.0
00060      PHI22=0.0
00061      DELTH=PI/36.
00062      CPHI=CPHI*RA5*DELTH
00063      C
00064      DO 30 I=1,18
00065      FI=I-1
00066      TH=FI*DELTH
00067      CTH4=COS(TH)**4
00068      AK3=A*SIN(TH)/COS(TH)
00069      AKNZ=AK3*ELT
00070      AKNZ2=AKNZ**2
00071      AKNT2=AKNY2+AKNZ2
00072      GAM11=AKNT2
00073      GAM12=-AKNX*AKNY/RSCAL
00074      GAM22=AKNX2/RSCAL2+AKNZ2*FPHIYY
00075      B2=EPS*AK1**2+AK2**2+AK3**2
00076      C
00077      IF (B2 .LT. 1.0E-05) THEN
00078      TERM11=CTH4*GAM11
00079      TERM12=CTH4*GAM12
00080      TERM22=CTH4*GAM22
00081      ELSE
00082      RB2=1.0/B2
00083      B4=B2**2
00084      RB4=1.0/B4
00085      TERM11=CTH4*RC2*GAM11*(1.0+AK1**2*OMEPS*RB2)**2
00086      TERM12=CTH4*RSQC*(GAM12+RB2*OMEPS*AK1*(GAM11*AK2+GAM12*AK1)
00087      &      +RB4*OMEPS2*AK1**3*AK2*GAM11)
00088      TERM22=CTH4*C*(GAM22+2.0*RB2*OMEPS*AK1*AK2*GAM12
00089      &      +RB4*OMEPS2*((AK1*AK2)**2)*GAM11)
00090      END IF
00091      C

```

```

00092         IF (I .LE. 1) THEN
00093             TERM11=0.5*TERM11
00094             TERM12=0.5*TERM12
00095             TERM22=0.5*TERM22
00096         END IF
00097     C
00098         PHI11=PHI11+TERM11
00099         PHI12=PHI12+TERM12
00100         PHI22=PHI22+TERM22
00101     30 END DO
00102     C
00103         PHIXX=PHI11*CPHI
00104         PHIXY=PHI12*CPHI
00105         PHIYY=PHI22*CPHI
00106         PHIXXO=PHIXX*CR2-TCRSR*PHIXY+PHIYY*SR2
00107         PHIXYO=(PHIXX-PHIYY)*CR*SR+PHIXY*(CR2-SR2)
00108         PHIYYO=PHIXX*SR2+TCRSR*PHIXY+PHIYY*CR2
00109     C
00110         RETURN
00111     END

00001     C
00002     C          EXINT          SUBROUTINE EXINT  -  EXPONENTIAL CURVE INTEGRATION
00003     C
00004     SUBROUTINE EXINT(Y1,Y2,X1,X2,YINT)
00005     C
00006     C
00007         YMAX=AMAX1(Y1,Y2)
00008         YMIN=AMIN1(Y1,Y2)
00009         DELX=ABS(X1-X2)
00010         YINT=YMAX*DELX
00011         IF (YMIN.EQ.00) YINT = 0.5*YINT
00012         IF (YMIN.EQ.00) GO TO 100
00013     C
00014         DELY=ALOG(YMAX/YMIN)
00015         IF (DELY .GE. 0.01) THEN
00016             COR1=(1.0-YMIN/YMAX)/DELY
00017             X  =SQRT(YMAX/YMIN-1.0)
00018             COR2=(ATAN(X))/X
00019             COR =SQRT(COR1*COR2)
00020             YINT=YINT*COR
00021         END IF
00022     C
00023     100 CONTINUE
00024         RETURN
00025     END

00001     C
00002     C          SGN          SIGN OF A FUNCTION OR PARAMETER
00003     C
00004     FUNCTION SGN(X)
00005     C
00006         IF (X .EQ. 0.) THEN
00007             SGN=0.
00008         ELSE
00009             SGN=X/ABS(X)
00010         END IF
00011     C

```

```

00012      RETURN
00013      END

00001  C
00002  C          FRESNL      FRESNEL INTEGRAL FUNCTIONS C(Z) AND S(Z)
00003  C
00004      SUBROUTINE FRESNL(Z,C,S)
00005  C
00006      PI=3.1415926
00007      X =0.5*PI*Z*Z
00008      COSX=COS(X)
00009      SINX=SIN(X)
00010  C
00011      TOP=1.0+0.926*Z
00012      BOT=2.0+Z*(1.792+3.104*Z)
00013      F  =TOP/BOT
00014  C
00015      TOP=1.0
00016      BOT=2.0+Z*(4.142+Z*(3.492+6.670*Z))
00017      G  =TOP/BOT
00018  C
00019      C  =  0.5 + F*SINX - G*COSX
00020      S  =  0.5 - F*COSX - G*SINX
00021  C
00022      RETURN
00023      END

00001  C
00002      SUBROUTINE TRANSROT ( PHI,M,AL,TR2,RE2 )
00003      REAL M,MT,MREL
00004      PI      = 3.14159265
00005      MT      = M*TAN(AL)
00006      MREL    = SQRT ( MT**2+M**2 )
00007      CPHI    = COS(PHI)
00008      SPHI    = SIN(PHI)
00009      DENPHI  = 1.+M**2+2.*M*CPHI
00010      CRIT    = ABS ( CPHI+M )
00011      CTH     = (-CPHI*(1.+M**2)-2.*M)/DENPHI
00012      STH     = (1.-M**2)*SPHI/DENPHI
00013      THETAD  = ATAN2(STH,CTH)*180./PI
00014      OPMCTH  = 1.+M*CTH
00015      STHPAL  = STH*COS(AL)+CTH*SIN(AL)
00016      RC      = -SIN(PHI+AL)/STHPAL
00017      RS      = -(COS(AL)-M*STH*SIN(AL)/OPMCTH)/STHPAL
00018      A11     = (1.+CPHI/M)
00019      A12     = (1.+CTH/M)*RS-STH/OPMCTH
00020      A21     = 1.+M*CPHI-MT*SPHI
00021      A22     = (OPMCTH-MT*STH)*RS-(M**2*STH/OPMCTH+MT)
00022      B1      = -(1.+CTH/M)*RC
00023      B2      = -(OPMCTH-MT*STH)*RC
00024      DET     = A11*A22-A12*A21
00025      OMT     = (A22*B1-A12*B2)/DET
00026      S       = (A11*B2-A21*B1)/DET
00027      R       = RC+RS*S
00028      TR      = 1.-OMT
00029      IF ( MREL.LT.1. ) TR2      = TR**2
00030      IF ( MREL.GE.1. ) TR2      = 0.000
00031      IF ( CRIT.GT.0.00 ) THEN

```

```

00032      RE2P    = R**2*(1.+M*CTH)*(CTH+M)/((1.+M*CPHI)*(CPHI+M))
00033      ELSE
00034      RE2P    = R**2*(1.+M*CTH)/(1.+M*CPHI)
00035      END IF
00036      IF ( MREL.LT.1. ) RE2    = ABS(RE2P)
00037      IF ( MREL.GE.1. ) RE2    = 1.000
00038      RETURN
00039      END

00001  C
00002      SUBROUTINE NEWSUB ( RLOW,RUPP,HTR,AKYN,IKY,NR,F3DB )
00003  C
00004      ord      = abs(akyn)*(1.+htr)/2.
00005  C
00006      N        = NR-1
00007      CALL SUB3D ( AKYN,IKY,HTR,RLOW,RUPP,N,F3DB )
00008      RETURN
00009      END

00001  C
00002      SUBROUTINE SUB3D ( KY,IKY,SIG,RL,RU,N,f3d1 )
00003      REAL KY,KMN,kmn2,kmns
00004      if ( (iky.eq.0).and.(n.eq.0) ) then
00005      f3d1 = (ru**2-rl**2)/(1.-sig**2)
00006      go to 100
00007      endif
00008      pi      = 3.14159265
00009      enc     = pi*float(n)
00010      kmn     = sqrt ( ky**2+(enc/(1.-sig))**2 )
00011      kmns    = kmn*sig
00012      kmn2    = kmn**2
00013      sig2    = sig**2
00014      ord     = abs(ky)*(1.+sig)/2.
00015      call simp ( rl,ru,kmn,sig,ord,cnn )
00016      call simpl ( kmn,sig,ord,dencn1 )
00017      f3d1    = ( 2.*cnn**2/dencn1 )/(ru**2-rl**2)
00018 100 continue
00019      return
00020      end

00001  c
00002      SUBROUTINE SIMP ( RL,RU,KMN,SIG,ORD,RES )
00003      REAL KMN
00004  c      # of points for Simpson's rule (minimum value 7:also needs to be odd)
00005  c      # of points determined assuming period of "2*pi."This must be
00006  c      reflected in rescaling of argument as below
00007      a      = kmn*rl
00008      b      = kmn*ru
00009      args   = kmn*sig
00010      factor = 1./kmn**2
00011  c      # of points for Simpson's rule (minimum value 7:also needs to be odd)
00012      pi     = 3.14159265
00013      rint   = (b-a)
00014      npml   = 5.*(rint/pi)
00015      np     = 2*(npml/2)+1
00016      if (np.lt.7) np = 7
00017      npml   = np-1
00018  c      evaluation by Simpson's rule with np points

```

```

00019      delx      = (b-a)/float(npml)
00020      c          multipliers for end,even and odd terms
00021      mulend     = 1.00
00022      muleven    = 4.00
00023      mulodd     = 2.00
00024      sum        = 0.00
00025      rargs      = args/ord
00026      if ((ord.le.30).or.(rargs.gt.0.9)) then
00027          call phijd ( args,ord,bjd )
00028          call phiryd ( args,ord,rbyd )
00029      endif
00030      if ((ord.gt.30).and.(rargs.le.0.9)) then
00031          call abesjd ( args,ord,bjd )
00032          call arbesyd ( args,ord,rbyd )
00033      endif
00034      do 200 i = 1,np
00035          x        = a+float(i-1)*delx
00036          rx       = x/ord
00037          if ((ord.le.30).or.(rx.gt.0.9)) then
00038              call phij ( x,ord,bj )
00039              call phiy ( x,ord,by )
00040              trm= by*rbyd
00041          endif
00042          if ((ord.gt.30).and.(rx.le.0.9)) then
00043              call abesj ( x,ord,bj )
00044              call abesyr ( x,args,ord,trm )
00045          endif
00046          y        = (bj-trm*bjd)*x
00047          if ((i.eq.1).or.(i.eq.np)) sum      = sum+y*mulend
00048          if ((i.ne.1).and.(i.ne.np)) then
00049              idisc = 2*(i/2)-i
00050              if (idisc.ne.0) sum      = sum+y*mulodd
00051              if (idisc.eq.0) sum      = sum+y*muleven
00052          endif
00053      200 continue
00054      aints      = delx*sum/3.
00055      res        = aints*factor
00056      return
00057      end

00001      c
00002      SUBROUTINE SIMP1 ( KMN,SIG,ORD,RES )
00003      REAL KMN
00004      c          # of points for Simpson's rule (minimum value 7:also needs to be odd)
00005      c          # of points determined assuming period of "2*pi."This must be
00006      c          reflected in rescaling of argument as below
00007      a          = kmn*sig
00008      b          = kmn
00009      args       = a
00010      factor     = 1./kmn**2
00011      c          # of points for Simpson's rule (minimum value 7:also needs to be odd)
00012      pi         = 3.14159265
00013      rint       = (b-a)
00014      npml       = 5.*(rint/pi)
00015      np         = 2*(npml/2)+1
00016      if (np.lt.7) np = 7
00017      npml       = np-1
00018      c          evaluation by Simpson's rule with np points
00019      delx       = (b-a)/float(npml)
00020      c          multipliers for end,even and odd terms

```

```

00021      mulend  = 1.00
00022      muleven = 4.00
00023      mulodd  = 2.00
00024      sum     = 0.00
00025      rargs   = args/ord
00026      if ((ord.le.30).or.(rargs.gt.0.9)) then
00027          call phijd ( args,ord,bjd )
00028          call phiryd ( args,ord,rbyd )
00029      endif
00030      if ((ord.gt.30).and.(rargs.le.0.9)) then
00031          call abesjd ( args,ord,bjd )
00032          call arbesyd ( args,ord,rbyd )
00033      endif
00034      do 200 i = 1,np
00035          x      = a+float(i-1)*delx
00036          rx     = x/ord
00037          if ((ord.le.30).or.(rx.gt.0.9)) then
00038              call phij ( x,ord,bj )
00039              call phiy ( x,ord,by )
00040              trm= by*rbyd
00041          endif
00042          if ((ord.gt.30).and.(rx.le.0.9)) then
00043              call abesj ( x,ord,bj )
00044              call abesyr ( x,args,ord,trm )
00045          endif
00046          y      = (bj-trm*bjd)**2*x
00047          if ((i.eq.1).or.(i.eq.np)) sum      = sum+y*mulend
00048          if ((i.ne.1).and.(i.ne.np)) then
00049              idisc = 2*(i/2)-i
00050              if (idisc.ne.0) sum      = sum+y*mulodd
00051              if (idisc.eq.0) sum      = sum+y*muleven
00052          endif
00053      200 continue
00054      aints  = delx*sum/3.
00055      res    = aints*factor
00056      return
00057      end

00001  c
00002      SUBROUTINE PHIJ ( ARG,ORD,BJARG )
00003      DIMENSION BJ(1000)
00004  C    CALCULATES " JORD(ARG) "
00005      NORD   = ORD
00006      NB     = NORD+1
00007      AL     = ORD-FLOAT(NORD)
00008      IF ( NB.GT.1000 ) PRINT *, 'NEED TO INCREASE DIMENSIONS OF',
00009  1      ' BJ IN SUBROUTINE PHIJ BEYOND 1000'
00010      IF ( NB.GT.1000 ) GO TO 200
00011  C    WRITE(12,*) 'RJBESL,ARG,ORD ',ARG,ORD
00012      CALL RJBESL ( ARG,AL,NB,BJ,NCALC )
00013      IF ( NCALC.LT.NB ) PRINT *, ' ERROR IN BJ CALCULATION !-PHIJ'
00014      IF ( NCALC.LT.NB ) PRINT *, 'ORD,ARG = ',ORD,ARG
00015      IF ( NCALC.LT.NB ) GO TO 100
00016      BJARG = BJ(NB)
00017  C    WRITE(12,*) 'RJBESL,RES ',BJARG
00018  100 CONTINUE
00019  200 CONTINUE
00020      return
00021      END

```

```

00001  C
00002      SUBROUTINE PHIY ( ARG,ORD,BYARG )
00003      DIMENSION BY(1000)
00004  C      CALCULATES " YORD(ARG) "
00005      NORD  = ORD
00006      NB    = NORD+1
00007      AL    = ORD-FLOAT(NORD)
00008  C      WRITE(12,*)'RYBESL,ARG,ORD ',ARG,ORD
00009      CALL RYBESL ( ARG,AL,NB,BY,NCALC )
00010      IF ( NCALC.LT.NB ) PRINT *, ' ERROR IN BY CALCULATION !-PHIY'
00011      IF ( NCALC.LT.NB ) PRINT *, 'ORD,ARG = ',ORD,ARG
00012      IF ( NCALC.LT.NB ) GO TO 100
00013      BYARG = BY(NB)
00014  C      WRITE(12,*)'RYBESL,RES ',BYARG
00015      100 CONTINUE
00016      200 CONTINUE
00017      return
00018      END

00001  C
00002      SUBROUTINE PHIJD ( ARG,ORD,BJDER )
00003      DIMENSION BJ(1000)
00004  C      CALCULATES " JORD'(ARG) "
00005      NORD  = ORD
00006      NB    = NORD+1
00007      NBP1  = NB+1
00008      IF ( NBP1.GT.1000 ) PRINT *, 'NEED TO INCREASE DIMENSIONS OF',
00009      1      ' BJ IN SUBROUTINE PHIJD BEYOND 1000'
00010      IF ( NBP1.GT.1000 ) GO TO 200
00011      AL    = ORD-FLOAT(NORD)
00012  C      WRITE(12,*)'RJBESL,ARG,ORD ',ARG,(ORD+1.)
00013      CALL RJBESL ( ARG,AL,NBP1,BJ,NCALC )
00014      IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN BJDER CALCULATION !'
00015      IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN PHIJD!'
00016      IF ( NCALC.LT.NBP1 ) PRINT *, 'ORD,ARG= ',ORD,ARG
00017      IF ( NCALC.LT.NBP1 ) GO TO 100
00018      BJDER = -BJ(NBP1)+ORD*BJ(NB)/ARG
00019  C      WRITE(12,*)'RJBESL,RES ',BJ(NBP1)
00020      100 CONTINUE
00021      200 CONTINUE
00022      return
00023      END

00001  C
00002      SUBROUTINE PHIRYD ( ARG,ORD,RBYDER )
00003      DIMENSION BY(1000)
00004  C      CALCULATES " JORD'(ARG) AND YORD'(ARG) "
00005      NORD  = ORD
00006      NB    = NORD+1
00007      NBP1  = NB+1
00008      IF ( NBP1.GT.1000 ) PRINT *, 'NEED TO INCREASE DIMENSIONS OF',
00009      1      ' BY IN SUBROUTINE PHIRYD BEYOND 1000'
00010      IF ( NBP1.GT.1000 ) GO TO 200
00011      AL    = ORD-FLOAT(NORD)
00012  C      WRITE(12,*)'RYBESL,ARG,ORD ',ARG,(ORD+1.)
00013      CALL RYBESL ( ARG,AL,NBP1,BY,NCALC )

```

```

00014         IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN BYDER CALCULATION ! '
00015         IF ( NCALC.LT.NBP1 ) PRINT *, ' ERROR IN PHIYD ! '
00016         IF ( NCALC.LT.NBP1 ) PRINT *, ' ORD, ARG = ', ORD, ARG
00017         IF ( NCALC.LT.NBP1 ) GO TO 100
00018         BYDER = -BY(NBP1)+ORD*BY(NB)/ARG
00019         RBYDER = 1./BYDER
00020 C      WRITE(12,*) 'RYBESL, RES ', BY(NBP1)
00021      100 CONTINUE
00022      200 CONTINUE
00023         return
00024         END

00001 C
00002         SUBROUTINE RJBESL(X, ALPHA, NB, B, NCALC)
00003 C-----
00004 C This routine calculates Bessel functions J sub(N+ALPHA) (X)
00005 C for non-negative argument X, and non-negative order N+ALPHA.
00006 C
00007 C
00008 C Explanation of variables in the calling sequence.
00009 C
00010 C X      - working precision non-negative real argument for which
00011 C          J's are to be calculated.
00012 C ALPHA - working precision fractional part of order for which
00013 C          J's or exponentially scaled J'r (J*exp(X)) are
00014 C          to be calculated. 0 <= ALPHA < 1.0.
00015 C NB    - integer number of functions to be calculated, NB > 0.
00016 C          The first function calculated is of order ALPHA, and the
00017 C          last is of order (NB - 1 + ALPHA).
00018 C B      - working precision output vector of length NB. If RJBESL
00019 C          terminates normally (NCALC=NB), the vector B contains the
00020 C          functions J/ALPHA/(X) through J/NB-1+ALPHA/(X), or the
00021 C          corresponding exponentially scaled functions.
00022 C NCALC - integer output variable indicating possible errors.
00023 C          Before using the vector B, the user should check that
00024 C          NCALC=NB, i.e., all orders have been calculated to
00025 C          the desired accuracy. See Error Returns below.
00026 C
00027 C
00028 C*****
00029 C*****
00030 C
00031 C Explanation of machine-dependent constants
00032 C
00033 C it      = Number of bits in the mantissa of a working precision
00034 C          variable
00035 C NSIG    = Decimal significance desired. Should be set to
00036 C          INT(LOG10(2)*it+1). Setting NSIG lower will result
00037 C          in decreased accuracy while setting NSIG higher will
00038 C          increase CPU time without increasing accuracy. The
00039 C          truncation error is limited to a relative error of
00040 C          T=.5*10**(-NSIG).
00041 C ENTEN   = 10.0 ** K, where K is the largest integer such that
00042 C          ENTEN is machine-representable in working precision
00043 C ENSIG    = 10.0 ** NSIG
00044 C RTNSIG   = 10.0 ** (-K) for the smallest integer K such that
00045 C          K .GE. NSIG/4
00046 C ENMTEN   = Smallest ABS(X) such that X/4 does not underflow
00047 C XLARGE   = Upper limit on the magnitude of X. If ABS(X)=N,

```

```

00048 C      then at least N iterations of the backward recursion
00049 C      will be executed.  The value of 10.0 ** 4 is used on
00050 C      every machine.
00051 C
00052 C
00053 C      Approximate values for some important machines are:
00054 C
00055 C
00056 C              it      NSIG      ENTEN      ENSIG
00057 C
00058 C      CRAY-1      (S.P.)      48      15      1.0E+2465      1.0E+15
00059 C      Cyber 180/855
00060 C      under NOS   (S.P.)      48      15      1.0E+322      1.0E+15
00061 C      IEEE (IBM/XT,
00062 C      SUN, etc.)  (S.P.)      24      8      1.0E+38      1.0E+8
00063 C      IEEE (IBM/XT,
00064 C      SUN, etc.)  (D.P.)      53      16      1.0D+308      1.0D+16
00065 C      IBM 3033     (D.P.)      14      5      1.0D+75      1.0D+5
00066 C      VAX          (S.P.)      24      8      1.0E+38      1.0E+8
00067 C      VAX D-Format (D.P.)      56      17      1.0D+38      1.0D+17
00068 C      VAX G-Format (D.P.)      53      16      1.0D+307      1.0D+16
00069 C
00070 C
00071 C              RTNSIG      ENMTEN      XLARGE
00072 C
00073 C      CRAY-1      (S.P.)      1.0E-4      1.84E-2466      1.0E+4
00074 C      Cyber 180/855
00075 C      under NOS   (S.P.)      1.0E-4      1.25E-293      1.0E+4
00076 C      IEEE (IBM/XT,
00077 C      SUN, etc.)  (S.P.)      1.0E-2      4.70E-38      1.0E+4
00078 C      IEEE (IBM/XT,
00079 C      SUN, etc.)  (D.P.)      1.0E-4      8.90D-308      1.0D+4
00080 C      IBM 3033     (D.P.)      1.0E-2      2.16D-78      1.0D+4
00081 C      VAX          (S.P.)      1.0E-2      1.17E-38      1.0E+4
00082 C      VAX D-Format (D.P.)      1.0E-5      1.17D-38      1.0D+4
00083 C      VAX G-Format (D.P.)      1.0E-4      2.22D-308      1.0D+4
00084 C
00085 C*****
00086 C*****
00087 C
00088 C      Error returns
00089 C
00090 C      In case of an error, NCALC .NE. NB, and not all J's are
00091 C      calculated to the desired accuracy.
00092 C
00093 C      NCALC .LT. 0:  An argument is out of range. For example,
00094 C      NBES .LE. 0, ALPHA .LT. 0 or .GT. 1, or X is too large.
00095 C      In this case, B(1) is set to zero, the remainder of the
00096 C      B-vector is not calculated, and NCALC is set to
00097 C      MIN(NB,0)-1 so that NCALC .NE. NB.
00098 C
00099 C      NB .GT. NCALC .GT. 0:  Not all requested function values could
00100 C      be calculated accurately. This usually occurs because NB is
00101 C      much larger than ABS(X). In this case, B(N) is calculated
00102 C      to the desired accuracy for N .LE. NCALC, but precision
00103 C      is lost for NCALC .LT. N .LE. NB. If B(N) does not vanish
00104 C      for N .GT. NCALC (because it is too small to be represented),
00105 C      and B(N)/B(NCALC) = 10**(-K), then only the first NSIG-K
00106 C      significant figures of B(N) can be trusted.
00107 C

```

```

00108 C
00109 C Intrinsic and other functions required are:
00110 C
00111 C     ABS, AINT, COS, DBLE, GAMMA (or DGAMMA), INT, MAX, MIN,
00112 C
00113 C     REAL, SIN, SQRT
00114 C
00115 C
00116 C Acknowledgement
00117 C
00118 C This program is based on a program written by David J. Sookne
00119 C (2) that computes values of the Bessel functions J or I of real
00120 C argument and integer order. Modifications include the restriction
00121 C of the computation to the J Bessel function of non-negative real
00122 C argument, the extension of the computation to arbitrary positive
00123 C order, and the elimination of most underflow.
00124 C
00125 C References: "A Note on Backward Recurrence Algorithms," Olver,
00126 C             F. W. J., and Sookne, D. J., Math. Comp. 26, 1972,
00127 C             pp 941-947.
00128 C
00129 C             "Bessel Functions of Real Argument and Integer Order,"
00130 C             Sookne, D. J., NBS Jour. of Res. B. 77B, 1973, pp
00131 C             125-132.
00132 C
00133 C Latest modification: March 19, 1990
00134 C
00135 C Author: W. J. Cody
00136 C         Applied Mathematics Division
00137 C         Argonne National Laboratory
00138 C         Argonne, IL 60439
00139 C
00140 C-----
00141 C     INTEGER I,J,K,L,M,MAGX,N,NB,NBMX,NCALC,NEND,NSTART
00142 C     REAL      GAMMA,
00143 CD    DOUBLE PRECISION  DGAMMA,
00144 C     1 ALPHA,ALPEM,ALP2EM,B,CAPP,CAPQ,CONV,EIGHTH,EM,EN,ENMTEN,ENSIG,
00145 C     2 ENTEN,FACT,FOUR,FUNC,GNU,HALF,HALFX,ONE,ONE30,P,PI2,PLAST,
00146 C     3 POLD,PSAVE,PSAVEL,RTNSIG,S,SUM,T,T1,TEMPA,TEMPB,TEMPC,TEST,
00147 C     4 THREE,THREE5,TOVER,TWO,TWOFIV,TWOPI1,TWOPI2,X,XC,XIN,XK,XLARGE,
00148 C     5 XM,VCOS,VSIN,Z,ZERO
00149 C     DIMENSION B(NB), FACT(25)
00150 C-----
00151 C Mathematical constants
00152 C
00153 C     PI2      - 2 / PI
00154 C     TWOPI1 - first few significant digits of 2 * PI
00155 C     TWOPI2 - (2*PI - TWOPI) to working precision, i.e.,
00156 C             TWOPI1 + TWOPI2 = 2 * PI to extra precision.
00157 C-----
00158 C     DATA PI2, TWOPI1, TWOPI2 /0.636619772367581343075535E0,6.28125E0,
00159 C     1 1.935307179586476925286767E-3/
00160 C     DATA ZERO, EIGHTH, HALF, ONE /0.0E0,0.125E0,0.5E0,1.0E0/
00161 C     DATA TWO, THREE, FOUR, TWOFIV /2.0E0,3.0E0,4.0E0,25.0E0/
00162 C     DATA ONE30, THREE5 /130.0E0,35.0E0/
00163 CD    DATA PI2, TWOPI1,TWOPI2/0.636619772367581343075535D0,6.28125D0,
00164 CD    1 1.935307179586476925286767D-3/
00165 CD    DATA ZERO, EIGHTH, HALF, ONE /0.0D0,0.125D0,0.5D0,1.0D0/
00166 CD    DATA TWO, THREE, FOUR, TWOFIV /2.0D0,3.0D0,4.0D0,25.0D0/
00167 CD    DATA ONE30, THREE5 /130.0D0,35.0D0/

```

```

00168 C-----
00169 C Machine-dependent parameters
00170 C-----
00171 DATA ENTEN, ENSIG, RTNSIG /1.0E38,1.0E8,1.0E-2/
00172 DATA ENMTEN, XLARGE /1.2E-37,1.0E4/
00173 CD DATA ENTEN, ENSIG, RTNSIG /1.0D38,1.0D17,1.0D-4/
00174 CD DATA ENMTEN, XLARGE /1.2D-37,1.0D4/
00175 C-----
00176 C Factorial(N)
00177 C-----
00178 DATA FACT /1.0E0,1.0E0,2.0E0,6.0E0,24.0E0,1.2E2,7.2E2,5.04E3,
00179 1 4.032E4,3.6288E5,3.6288E6,3.99168E7,4.790016E8,6.2270208E9,
00180 2 8.71782912E10,1.307674368E12,2.0922789888E13,3.55687428096E14,
00181 3 6.402373705728E15,1.21645100408832E17,2.43290200817664E18,
00182 4 5.109094217170944E19,1.12400072777760768E21,
00183 5 2.585201673888497664E22,6.2044840173323943936E23/
00184 CD DATA FACT /1.0D0,1.0D0,2.0D0,6.0D0,24.0D0,1.2D2,7.2D2,5.04D3,
00185 CD 1 4.032D4,3.6288D5,3.6288D6,3.99168D7,4.790016D8,6.2270208D9,
00186 CD 2 8.71782912D10,1.307674368D12,2.0922789888D13,3.55687428096D14,
00187 CD 3 6.402373705728D15,1.21645100408832D17,2.43290200817664D18,
00188 CD 4 5.109094217170944D19,1.12400072777760768D21,
00189 CD 5 2.585201673888497664D22,6.2044840173323943936D23/
00190 C-----
00191 C Statement functions for conversion and the gamma function.
00192 C-----
00193 CONV(I) = REAL(I)
00194 FUNC(X) = GAMMA(X)
00195 CD CONV(I) = DBLE(I)
00196 CD FUNC(X) = DGAMMA(X)
00197 C-----
00198 C Check for out of range arguments.
00199 C-----
00200 MAGX = INT(X)
00201 IF ((NB.GT.0) .AND. (X.GE.ZERO) .AND. (X.LE.XLARGE)
00202 1 .AND. (ALPHA.GE.ZERO) .AND. (ALPHA.LT.ONE))
00203 2 THEN
00204 C-----
00205 C Initialize result array to zero.
00206 C-----
00207 NCALC = NB
00208 DO 20 I=1,NB
00209 B(I) = ZERO
00210 20 CONTINUE
00211 C-----
00212 C Branch to use 2-term ascending series for small X and asymptotic
00213 C form for large X when NB is not too large.
00214 C-----
00215 IF (X.LT.RTNSIG) THEN
00216 C-----
00217 C Two-term ascending series for small X.
00218 C-----
00219 TEMPA = ONE
00220 ALPEM = ONE + ALPHA
00221 HALFX = ZERO
00222 IF (X.GT.ENMTEN) HALFX = HALF*X
00223 IF (ALPHA.NE.ZERO)
00224 1 TEMPA = HALFX**ALPHA/(ALPHA*FUNC(ALPHA))
00225 TEMPB = ZERO
00226 IF ((X+ONE).GT.ONE) TEMPB = -HALFX*HALFX
00227 B(1) = TEMPA + TEMPA*TEMPB/ALPEM

```

```

00228             IF ((X.NE.ZERO) .AND. (B(1).EQ.ZERO)) NCALC = 0
00229             IF (NB .NE. 1) THEN
00230                 IF (X .LE. ZERO) THEN
00231                     DO 30 N=2,NB
00232                         B(N) = ZERO
00233             30             CONTINUE
00234                 ELSE
00235             C-----
00236             C Calculate higher order functions.
00237             C-----
00238                     TEMPC = HALFX
00239                     TOVER = (ENMTEN+ENMTEN)/X
00240                     IF (TEMPB.NE.ZERO) TOVER = ENMTEN/TEMPB
00241                     DO 50 N=2,NB
00242                         TEMPA = TEMPA/ALPEM
00243                         ALPEM = ALPEM + ONE
00244                         TEMPA = TEMPA*TEMPC
00245                         IF (TEMPA.LE.TOVER*ALPEM) TEMPA = ZERO
00246                         B(N) = TEMPA + TEMPA*TEMPB/ALPEM
00247                         IF ((B(N).EQ.ZERO) .AND. (NCALC.GT.N))
00248             1             NCALC = N-1
00249             50             CONTINUE
00250                     END IF
00251             END IF
00252             ELSE IF ((X.GT.TWOFIV) .AND. (NB.LE.MAGX+1)) THEN
00253             C-----
00254             C Asymptotic series for X .GT. 21.0.
00255             C-----
00256                     XC = SQRT(PI2/X)
00257                     XIN = (EIGHTH/X)**2
00258                     M = 11
00259                     IF (X.GE.THREE5) M = 8
00260                     IF (X.GE.ONE30) M = 4
00261                     XM = FOUR*CONV(M)
00262             C-----
00263             C Argument reduction for SIN and COS routines.
00264             C-----
00265                     T = AINT(X/(TWOPI1+TWOPI2)+HALF)
00266                     Z = ((X-T*TWOPI1)-T*TWOPI2) - (ALPHA+HALF)/PI2
00267                     VSIN = SIN(Z)
00268                     VCOS = COS(Z)
00269                     GNU = ALPHA + ALPHA
00270                     DO 80 I=1,2
00271                         S = ((XM-ONE)-GNU)*((XM-ONE)+GNU)*XIN*HALF
00272                         T = (GNU-(XM-THREE))*(GNU+(XM-THREE))
00273                         CAPP = S*T/FACT(2*M+1)
00274                         T1 = (GNU-(XM+ONE))*(GNU+(XM+ONE))
00275                         CAPQ = S*T1/FACT(2*M+2)
00276                         XK = XM
00277                         K = M + M
00278                         T1 = T
00279                     DO 70 J=2,M
00280                         XK = XK - FOUR
00281                         S = ((XK-ONE)-GNU)*((XK-ONE)+GNU)
00282                         T = (GNU-(XK-THREE))*(GNU+(XK-THREE))
00283                         CAPP = (CAPP+ONE/FACT(K-1))*S*T*XIN
00284                         CAPQ = (CAPQ+ONE/FACT(K))*S*T1*XIN
00285                         K = K - 2
00286                         T1 = T
00287             70             CONTINUE

```

```

00288          CAPP = CAPP + ONE
00289          CAPQ = (CAPQ+ONE)*(GNU*GNU-ONE)*(EIGHTH/X)
00290          B(I) = XC*(CAPP*VCOS-CAPQ*VSIN)
00291          IF (NB.EQ.1) GO TO 300
00292          T = VSIN
00293          VSIN = -VCOS
00294          VCOS = T
00295          GNU = GNU + TWO
00296      80          CONTINUE
00297      C-----
00298      C If NB .GT. 2, compute J(X,ORDER+I)  I = 2, NB-1
00299      C-----
00300          IF (NB .GT. 2) THEN
00301              GNU = ALPHA + ALPHA + TWO
00302              DO 90 J=3,NB
00303                  B(J) = GNU*B(J-1)/X - B(J-2)
00304                  GNU = GNU + TWO
00305      90          CONTINUE
00306          END IF
00307      C-----
00308      C Use recurrence to generate results.  First initialize the
00309      C calculation of P*S.
00310      C-----
00311          ELSE
00312              NBMX = NB - MAGX
00313              N = MAGX + 1
00314              EN = CONV(N+N) + (ALPHA+ALPHA)
00315              PLAST = ONE
00316              P = EN/X
00317      C-----
00318      C Calculate general significance test.
00319      C-----
00320          TEST = ENSIG + ENSIG
00321          IF (NBMX .GE. 3) THEN
00322      C-----
00323      C Calculate P*S until N = NB-1.  Check for possible overflow.
00324      C-----
00325              TOVER = ENTEN/ENSIG
00326              NSTART = MAGX + 2
00327              NEND = NB - 1
00328              EN = CONV(NSTART+NSTART) - TWO + (ALPHA+ALPHA)
00329              DO 130 K=NSTART,NEND
00330                  N = K
00331                  EN = EN + TWO
00332                  POLD = PLAST
00333                  PLAST = P
00334                  P = EN*PLAST/X - POLD
00335                  IF (P.GT.TOVER) THEN
00336      C-----
00337      C To avoid overflow, divide P*S by TOVER.  Calculate P*S until
00338      C ABS(P) .GT. 1.
00339      C-----
00340                  TOVER = ENTEN
00341                  P = P/TOVER
00342                  PLAST = PLAST/TOVER
00343                  PSAVE = P
00344                  PSAVEL = PLAST
00345                  NSTART = N + 1
00346      100         N = N + 1
00347                  EN = EN + TWO

```

```

00348             POLD = PLAST
00349             PLAST = P
00350             P = EN*PLAST/X - POLD
00351             IF (P.LE.ONE) GO TO 100
00352             TEMPB = EN/X
00353 C-----
00354 C Calculate backward test and find NCALC, the highest N such that
00355 C the test is passed.
00356 C-----
00357             TEST = POLD*PLAST*(HALF-HALF/(TEMPB*TEMPB))
00358             TEST = TEST/ENSIG
00359             P = PLAST*TOVER
00360             N = N - 1
00361             EN = EN - TWO
00362             NEND = MIN(NB,N)
00363             DO 110 L=NSTART,NEND
00364                 POLD = PSAVE
00365                 PSAVE = PSAVE
00366                 PSAVE = EN*PSAVE/X - POLD
00367                 IF (PSAVE*PSAVE.GT.TEST) THEN
00368                     NCALC = L - 1
00369                     GO TO 190
00370                 END IF
00371 110             CONTINUE
00372                 NCALC = NEND
00373                 GO TO 190
00374             END IF
00375 130             CONTINUE
00376                 N = NEND
00377                 EN = CONV(N+N) + (ALPHA+ALPHA)
00378 C-----
00379 C Calculate special significance test for NBMX .GT. 2.
00380 C-----
00381             TEST = MAX(TEST,SQRT(PLAST*ENSIG)*SQRT(P+P))
00382             END IF
00383 C-----
00384 C Calculate P*S until significance test passes.
00385 C-----
00386 140             N = N + 1
00387                 EN = EN + TWO
00388                 POLD = PLAST
00389                 PLAST = P
00390                 P = EN*PLAST/X - POLD
00391                 IF (P.LT.TEST) GO TO 140
00392 C-----
00393 C Initialize the backward recursion and the normalization sum.
00394 C-----
00395 190             N = N + 1
00396                 EN = EN + TWO
00397                 TEMPB = ZERO
00398                 TEMPA = ONE/P
00399                 M = 2*N - 4*(N/2)
00400                 SUM = ZERO
00401                 EM = CONV(N/2)
00402                 ALPEM = (EM-ONE) + ALPHA
00403                 ALP2EM = (EM+EM) + ALPHA
00404                 IF (M .NE. 0) SUM = TEMPA*ALPEM*ALP2EM/EM
00405                 NEND = N - NB
00406                 IF (NEND .GT. 0) THEN
00407 C-----

```

```

00408 C Recur backward via difference equation, calculating (but not
00409 C storing) B(N), until N = NB.
00410 C-----
00411 DO 200 L=1,NEND
00412 N = N - 1
00413 EN = EN - TWO
00414 TEMPC = TEMPB
00415 TEMPB = TEMPA
00416 TEMPA = (EN*TEMPB)/X - TEMPC
00417 M = 2 - M
00418 IF (M .NE. 0) THEN
00419 EM = EM - ONE
00420 ALP2EM = (EM+EM) + ALPHA
00421 IF (N.EQ.1) GO TO 210
00422 ALPEM = (EM-ONE) + ALPHA
00423 IF (ALPEM.EQ.ZERO) ALPEM = ONE
00424 SUM = (SUM+TEMPA*ALP2EM)*ALPEM/EM
00425 END IF
00426 200 CONTINUE
00427 END IF
00428 C-----
00429 C Store B(NB).
00430 C-----
00431 210 B(N) = TEMPA
00432 IF (NEND .GE. 0) THEN
00433 IF (NB .LE. 1) THEN
00434 ALP2EM = ALPHA
00435 IF ((ALPHA+ONE).EQ.ONE) ALP2EM = ONE
00436 SUM = SUM + B(1)*ALP2EM
00437 GO TO 250
00438 ELSE
00439 C-----
00440 C Calculate and store B(NB-1).
00441 C-----
00442 N = N - 1
00443 EN = EN - TWO
00444 B(N) = (EN*TEMPA)/X - TEMPB
00445 IF (N.EQ.1) GO TO 240
00446 M = 2 - M
00447 IF (M .NE. 0) THEN
00448 EM = EM - ONE
00449 ALP2EM = (EM+EM) + ALPHA
00450 ALPEM = (EM-ONE) + ALPHA
00451 IF (ALPEM.EQ.ZERO) ALPEM = ONE
00452 SUM = (SUM+B(N)*ALP2EM)*ALPEM/EM
00453 END IF
00454 END IF
00455 END IF
00456 NEND = N - 2
00457 IF (NEND .NE. 0) THEN
00458 C-----
00459 C Calculate via difference equation and store B(N), until N = 2.
00460 C-----
00461 DO 230 L=1,NEND
00462 N = N - 1
00463 EN = EN - TWO
00464 B(N) = (EN*B(N+1))/X - B(N+2)
00465 M = 2 - M
00466 IF (M .NE. 0) THEN
00467 EM = EM - ONE

```

```

00468             ALP2EM = (EM+EM) + ALPHA
00469             ALPEM = (EM-ONE) + ALPHA
00470             IF (ALPEM.EQ.ZERO) ALPEM = ONE
00471             SUM = (SUM+B(N)*ALP2EM)*ALPEM/EM
00472             END IF
00473     230             CONTINUE
00474             END IF
00475 C-----
00476 C Calculate B(1).
00477 C-----
00478             B(1) = TWO*(ALPHA+ONE)*B(2)/X - B(3)
00479     240             EM = EM - ONE
00480             ALP2EM = (EM+EM) + ALPHA
00481             IF (ALP2EM.EQ.ZERO) ALP2EM = ONE
00482             SUM = SUM + B(1)*ALP2EM
00483 C-----
00484 C Normalize. Divide all B(N) by sum.
00485 C-----
00486     250             IF ((ALPHA+ONE).NE.ONE)
00487     1             SUM = SUM*FUNC(ALPHA)*(X*HALF)**(-ALPHA)
00488             TEMPA = ENMTEN
00489             IF (SUM.GT.ONE) TEMPA = TEMPA*SUM
00490             DO 260 N=1,NB
00491                 IF (ABS(B(N)).LT.TEMPA) B(N) = ZERO
00492                 B(N) = B(N)/SUM
00493     260             CONTINUE
00494             END IF
00495 C-----
00496 C Error return -- X, NB, or ALPHA is out of range.
00497 C-----
00498             ELSE
00499                 B(1) = ZERO
00500                 NCALC = MIN(NB,0) - 1
00501             END IF
00502 C-----
00503 C Exit
00504 C-----
00505     300 RETURN
00506 C ----- Last line of RJBESL -----
00507             END

00001
00002
00003
00004
00005
00006             SUBROUTINE RYBESL(X,ALPHA,NB,BY,NCALC)
00007 C-----
00008 C
00009 C This routine calculates Bessel functions Y SUB(N+ALPHA) (X)
00010 C for non-negative argument X, and non-negative order N+ALPHA.
00011 C
00012 C
00013 C Explanation of variables in the calling sequence
00014 C
00015 C X      - Working precision non-negative real argument for which
00016 C          Y's are to be calculated.
00017 C ALPHA  - Working precision fractional part of order for which
00018 C          Y's are to be calculated. 0 .LE. ALPHA .LT. 1.0.
00019 C NB     - Integer number of functions to be calculated, NB .GT. 0.

```

```

00020 C      The first function calculated is of order ALPHA, and the
00021 C      last is of order (NB - 1 + ALPHA).
00022 C BY    - Working precision output vector of length NB.  If the
00023 C      routine terminates normally (NCALC=NB), the vector BY
00024 C      contains the functions Y(ALPHA,X), ... , Y(NB-1+ALPHA,X),
00025 C      If (0 .LT. NCALC .LT. NB), BY(I) contains correct function
00026 C      values for I .LE. NCALC, and contains the ratios
00027 C      Y(ALPHA+I-1,X)/Y(ALPHA+I-2,X) for the rest of the array.
00028 C NCALC - Integer output variable indicating possible errors.
00029 C      Before using the vector BY, the user should check that
00030 C      NCALC=NB, i.e., all orders have been calculated to
00031 C      the desired accuracy.  See error returns below.
00032 C
00033 C
00034 C*****
00035 C*****
00036 C
00037 C Explanation of machine-dependent constants
00038 C
00039 C      beta  = Radix for the floating-point system
00040 C      p      = Number of significant base-beta digits in the
00041 C              significand of a floating-point number
00042 C      minexp = Smallest representable power of beta
00043 C      maxexp = Smallest power of beta that overflows
00044 C      EPS    = beta ** (-p)
00045 C      DEL    = Machine number below which sin(x)/x = 1; approximately
00046 C              Sqrt(EPS).
00047 C      XMIN   = Smallest acceptable argument for RBESY; approximately
00048 C              max(2*beta**minexp,2/XINF), rounded up
00049 C      XINF   = Largest positive machine number; approximately
00050 C              beta**maxexp
00051 C      THRESH = Lower bound for use of the asymptotic form; approximately
00052 C              Aint(-Log10(EPS/2.0))+1.0
00053 C      XLARGE = Upper bound on X; approximately 1/DEL, because the sine
00054 C              and cosine functions have lost about half of their
00055 C              precision at that point.
00056 C
00057 C
00058 C      Approximate values for some important machines are:
00059 C
00060 C              beta      p      minexp      maxexp      EPS
00061 C
00062 C CRAY-1      (S.P.)      2      48      -8193      8191      3.55E-15
00063 C Cyber 180/185
00064 C   under NOS  (S.P.)      2      48      -975      1070      3.55E-15
00065 C IEEE (IBM/XT,
00066 C   SUN, etc.) (S.P.)      2      24      -126      128      5.96E-8
00067 C IEEE (IBM/XT,
00068 C   SUN, etc.) (D.P.)      2      53      -1022     1024      1.11D-16
00069 C IBM 3033     (D.P.)     16      14        -65         63      1.39D-17
00070 C VAX          (S.P.)      2      24      -128      127      5.96E-8
00071 C VAX D-Format (D.P.)      2      56      -128      127      1.39D-17
00072 C VAX G-Format (D.P.)      2      53     -1024     1023      1.11D-16
00073 C
00074 C
00075 C              DEL      XMIN      XINF      THRESH  XLARGE
00076 C
00077 C CRAY-1      (S.P.)    5.0E-8    3.67E-2466  5.45E+2465  15.0E0    2.0E7
00078 C Cyber 180/855
00079 C   under NOS  (S.P.)    5.0E-8    6.28E-294   1.26E+322   15.0E0    2.0E7

```

```

00080 C IEEE (IBM/XT,
00081 C   SUN, etc.) (S.P.) 1.0E-4 2.36E-38 3.40E+38 8.0E0 1.0E4
00082 C IEEE (IBM/XT,
00083 C   SUN, etc.) (D.P.) 1.0D-8 4.46D-308 1.79D+308 16.0D0 1.0D8
00084 C IBM 3033 (D.P.) 1.0D-8 2.77D-76 7.23D+75 17.0D0 1.0D8
00085 C VAX (S.P.) 1.0E-4 1.18E-38 1.70E+38 8.0E0 1.0E4
00086 C VAX D-Format (D.P.) 1.0D-9 1.18D-38 1.70D+38 17.0D0 1.0D9
00087 C VAX G-Format (D.P.) 1.0D-8 2.23D-308 8.98D+307 16.0D0 1.0D8
00088 C
00089 C*****
00090 C*****
00091 C
00092 C Error returns
00093 C
00094 C In case of an error, NCALC .NE. NB, and not all Y's are
00095 C calculated to the desired accuracy.
00096 C
00097 C NCALC .LT. -1: An argument is out of range. For example,
00098 C   NB .LE. 0, IZE is not 1 or 2, or IZE=1 and ABS(X) .GE.
00099 C   XMAX. In this case, BY(1) = 0.0, the remainder of the
00100 C   BY-vector is not calculated, and NCALC is set to
00101 C   MIN0(NB,0)-2 so that NCALC .NE. NB.
00102 C NCALC = -1: Y(ALPHA,X) .GE. XINF. The requested function
00103 C values are set to 0.0.
00104 C 1 .LT. NCALC .LT. NB: Not all requested function values could
00105 C be calculated accurately. BY(I) contains correct function
00106 C values for I .LE. NCALC, and the remaining NB-NCALC
00107 C array elements contain 0.0.
00108 C
00109 C
00110 C Intrinsic functions required are:
00111 C
00112 C   DBLE, EXP, INT, MAX, MIN, REAL, SQRT
00113 C
00114 C
00115 C Acknowledgement
00116 C
00117 C This program draws heavily on Temme's Algol program for Y(a,x)
00118 C and Y(a+1,x) and on Campbell's programs for Y_nu(x). Temme's
00119 C scheme is used for x < THRESH, and Campbell's scheme is used
00120 C in the asymptotic region. Segments of code from both sources
00121 C have been translated into Fortran 77, merged, and heavily modified.
00122 C Modifications include parameterization of machine dependencies,
00123 C use of a new approximation for ln(gamma(x)), and built-in
00124 C protection against over/underflow.
00125 C
00126 C References: "Bessel functions J_nu(x) and Y_nu(x) of real
00127 C order and real argument," Campbell, J. B.,
00128 C Comp. Phy. Comm. 18, 1979, pp. 133-142.
00129 C
00130 C "On the numerical evaluation of the ordinary
00131 C Bessel function of the second kind," Temme,
00132 C N. M., J. Comput. Phys. 21, 1976, pp. 343-350.
00133 C
00134 C Latest modification: March 19, 1990
00135 C
00136 C Modified by: W. J. Cody
00137 C   Applied Mathematics Division
00138 C   Argonne National Laboratory
00139 C   Argonne, IL 60439

```

```

00140 C
00141 C-----
00142     INTEGER I,K,NA,NB,NCALC
00143     REAL
00144 CD     DOUBLE PRECISION
00145     1  ALFA,ALPHA,AYE,B,BY,C,CH,COSMU,D,DEL,DEN,DDIV,DIV,DMU,D1,D2,
00146     2  E,EIGHT,EN,ENU,EN1,EPS,EVEN,EX,F,FIVPI,G,GAMMA,H,HALF,ODD,
00147     3  ONBPI,ONE,ONE5,P,PA,PA1,PI,PIBY2,PIM5,Q,QA,QA1,Q0,R,S,SINMU,
00148     4  SQ2BPI,TEN9,TERM,THREE,THRESH,TWO,TWOBYX,X,XINF,XLARGE,XMIN,
00149     5  XNA,X2,YA,YA1,ZERO
00150     DIMENSION BY(NB),CH(21)
00151 C-----
00152 C  Mathematical constants
00153 C    FIVPI = 5*PI
00154 C    PIM5 = 5*PI - 15
00155 C    ONBPI = 1/PI
00156 C    PIBY2 = PI/2
00157 C    SQ2BPI = SQUARE ROOT OF 2/PI
00158 C-----
00159     DATA ZERO,HALF,ONE,TWO,THREE/0.0E0,0.5E0,1.0E0,2.0E0,3.0E0/
00160     DATA EIGHT,ONE5,TEN9/8.0E0,15.0E0,1.9E1/
00161     DATA FIVPI,PIBY2/1.5707963267948966192E1,1.5707963267948966192E0/
00162     DATA PI,SQ2BPI/3.1415926535897932385E0,7.9788456080286535588E-1/
00163     DATA PIM5,ONBPI/7.0796326794896619231E-1,3.1830988618379067154E-1/
00164     CD  DATA ZERO,HALF,ONE,TWO,THREE/0.0D0,0.5D0,1.0D0,2.0D0,3.0D0/
00165     CD  DATA EIGHT,ONE5,TEN9/8.0D0,15.0D0,1.9D1/
00166     CD  DATA FIVPI,PIBY2/1.5707963267948966192D1,1.5707963267948966192D0/
00167     CD  DATA PI,SQ2BPI/3.1415926535897932385D0,7.9788456080286535588D-1/
00168     CD  DATA PIM5,ONBPI/7.0796326794896619231D-1,3.1830988618379067154D-1/
00169 C-----
00170 C  Machine-dependent constants
00171 C-----
00172     DATA DEL,XMIN,XINF,EPS/1.0E-4,2.36E-38,1.70E38,5.96E-8/
00173     DATA THRESH,XLARGE/8.0E0,1.0E4/
00174 CD  DATA DEL,XMIN,XINF,EPS/1.0D-8,4.46D-308,1.79D308,1.11D-16/
00175 CD  DATA THRESH,XLARGE/16.0D0,1.0D8/
00176 C-----
00177 C  Coefficients for Chebyshev polynomial expansion of
00178 C    1/gamma(1-x), abs(x) .le. .5
00179 C-----
00180     DATA CH/-0.67735241822398840964E-23,-0.61455180116049879894E-22,
00181     1      0.29017595056104745456E-20, 0.13639417919073099464E-18,
00182     2      0.23826220476859635824E-17,-0.90642907957550702534E-17,
00183     3      -0.14943667065169001769E-14,-0.33919078305362211264E-13,
00184     4      -0.17023776642512729175E-12, 0.91609750938768647911E-11,
00185     5      0.24230957900482704055E-09, 0.17451364971382984243E-08,
00186     6      -0.33126119768180852711E-07,-0.86592079961391259661E-06,
00187     7      -0.49717367041957398581E-05, 0.76309597585908126618E-04,
00188     8      0.12719271366545622927E-02, 0.17063050710955562222E-02,
00189     9      -0.76852840844786673690E-01,-0.28387654227602353814E+00,
00190     A      0.92187029365045265648E+00/
00191 CD  DATA CH/-0.67735241822398840964D-23,-0.61455180116049879894D-22,
00192 CD  1      0.29017595056104745456D-20, 0.13639417919073099464D-18,
00193 CD  2      0.23826220476859635824D-17,-0.90642907957550702534D-17,
00194 CD  3      -0.14943667065169001769D-14,-0.33919078305362211264D-13,
00195 CD  4      -0.17023776642512729175D-12, 0.91609750938768647911D-11,
00196 CD  5      0.24230957900482704055D-09, 0.17451364971382984243D-08,
00197 CD  6      -0.33126119768180852711D-07,-0.86592079961391259661D-06,
00198 CD  7      -0.49717367041957398581D-05, 0.76309597585908126618D-04,
00199 CD  8      0.12719271366545622927D-02, 0.17063050710955562222D-02,

```

```

00200  CD   9      -0.76852840844786673690D-01,-0.28387654227602353814D+00,
00201  CD   A      0.92187029365045265648D+00/
00202  C-----
00203      EX = X
00204      ENU = ALPHA
00205      IF ((NB .GT. 0) .AND. (X .GE. XMIN) .AND. (EX .LT. XLARGE)
00206  1      .AND. (ENU .GE. ZERO) .AND. (ENU .LT. ONE)) THEN
00207          XNA = AINT(ENU+HALF)
00208          NA = INT(XNA)
00209          IF (NA .EQ. 1) ENU = ENU - XNA
00210          IF (ENU .EQ. -HALF) THEN
00211              P = SQ2BPPI/SQRT(EX)
00212              YA = P * SIN(EX)
00213              YA1 = -P * COS(EX)
00214              ELSE IF (EX .LT. THREE) THEN
00215  C-----
00216  C   Use Temme's scheme for small X
00217  C-----
00218              B = EX * HALF
00219              D = -LOG(B)
00220              F = ENU * D
00221              E = B**(-ENU)
00222              IF (ABS(ENU) .LT. DEL) THEN
00223                  C = ONBPPI
00224              ELSE
00225                  C = ENU / SIN(ENU*PI)
00226              END IF
00227  C-----
00228  C   Computation of sinh(f)/f
00229  C-----
00230              IF (ABS(F) .LT. ONE) THEN
00231                  X2 = F*F
00232                  EN = TEN9
00233                  S = ONE
00234                  DO 80 I = 1, 9
00235                      S = S*X2/EN/(EN-ONE)+ONE
00236                      EN = EN - TWO
00237  80                  CONTINUE
00238              ELSE
00239                  S = (E - ONE/E) * HALF / F
00240              END IF
00241  C-----
00242  C   Computation of 1/gamma(1-a) using Chebyshev polynomials
00243  C-----
00244              X2 = ENU*ENU*EIGHT
00245              AYE = CH(1)
00246              EVEN = ZERO
00247              ALFA = CH(2)
00248              ODD = ZERO
00249              DO 40 I = 3, 19, 2
00250                  EVEN = -(AYE+AYE+EVEN)
00251                  AYE = -EVEN*X2 - AYE + CH(I)
00252                  ODD = -(ALFA+ALFA+ODD)
00253                  ALFA = -ODD*X2 - ALFA + CH(I+1)
00254  40                  CONTINUE
00255                  EVEN = (EVEN*HALF+AYE)*X2 - AYE + CH(21)
00256                  ODD = (ODD+ALFA)*TWO
00257                  GAMMA = ODD*ENU + EVEN
00258  C-----
00259  C   End of computation of 1/gamma(1-a)

```

```

00260 C-----
00261      G = E * GAMMA
00262      E = (E + ONE/E) * HALF
00263      F = TWO*C*(ODD*E+EVEN*S*D)
00264      E = ENU*ENU
00265      P = G*C
00266      Q = ONBPI / G
00267      C = ENU*PIBY2
00268      IF (ABS(C) .LT. DEL) THEN
00269          R = ONE
00270      ELSE
00271          R = SIN(C)/C
00272      END IF
00273      R = PI*C*R*R
00274      C = ONE
00275      D = - B*B
00276      H = ZERO
00277      YA = F + R*Q
00278      YA1 = P
00279      EN = ZERO
00280      100      EN = EN + ONE
00281      IF (ABS(G/(ONE+ABS(YA)))
00282          1      + ABS(H/(ONE+ABS(YA1))) .GT. EPS) THEN
00283          F = (F*EN+P+Q)/(EN*EN-E)
00284          C = C * D/EN
00285          P = P/(EN-ENU)
00286          Q = Q/(EN+ENU)
00287          G = C*(F+R*Q)
00288          H = C*P - EN*G
00289          YA = YA + G
00290          YA1 = YA1+H
00291          GO TO 100
00292      END IF
00293      YA = -YA
00294      YA1 = -YA1/B
00295      ELSE IF (EX .LT. THRESH) THEN
00296 C-----
00297 C Use Temme's scheme for moderate X
00298 C-----
00299      C = (HALF-ENU)*(HALF+ENU)
00300      B = EX + EX
00301      E = (EX*ONBPI*COS(ENU*PI)/EPS)
00302      E = E*E
00303      P = ONE
00304      Q = -EX
00305      R = ONE + EX*EX
00306      S = R
00307      EN = TWO
00308      200      IF (R*EN*EN .LT. E) THEN
00309          EN1 = EN+ONE
00310          D = (EN-ONE+C/EN)/S
00311          P = (EN+EN-P*D)/EN1
00312          Q = (-B+Q*D)/EN1
00313          S = P*P + Q*Q
00314          R = R*S
00315          EN = EN1
00316          GO TO 200
00317      END IF
00318      F = P/S
00319      P = F

```

```

00320          G = -Q/S
00321          Q = G
00322          220      EN = EN - ONE
00323          IF (EN .GT. ZERO) THEN
00324              R = EN1*(TWO-P)-TWO
00325              S = B + EN1*Q
00326              D = (EN-ONE+C/EN)/(R*R+S*S)
00327              P = D*R
00328              Q = D*S
00329              E = F + ONE
00330              F = P*E - G*Q
00331              G = Q*E + P*G
00332              EN1 = EN
00333              GO TO 220
00334          END IF
00335          F = ONE + F
00336          D = F*F + G*G
00337          PA = F/D
00338          QA = -G/D
00339          D = ENU + HALF -P
00340          Q = Q + EX
00341          PA1 = (PA*Q-QA*D)/EX
00342          QA1 = (QA*Q+PA*D)/EX
00343          B = EX - PIBY2*(ENU+HALF)
00344          C = COS(B)
00345          S = SIN(B)
00346          D = SQ2BPI/SQRT(EX)
00347          YA = D*(PA*S+QA*C)
00348          YA1 = D*(QA1*S-PA1*C)
00349          ELSE
00350      C-----
00351      C  Use Campbell's asymptotic scheme.
00352      C-----
00353          NA = 0
00354          D1 = AINT(EX/FIVPI)
00355          I = INT(D1)
00356          DMU = ((EX-ONE5*D1)-D1*PIM5)-(ALPHA+HALF)*PIBY2
00357          IF (I-2*(I/2) .EQ. 0) THEN
00358              COSMU = COS(DMU)
00359              SINMU = SIN(DMU)
00360          ELSE
00361              COSMU = -COS(DMU)
00362              SINMU = -SIN(DMU)
00363          END IF
00364          DDIV = EIGHT * EX
00365          DMU = ALPHA
00366          DEN = SQRT(EX)
00367          DO 350 K = 1, 2
00368              P = COSMU
00369              COSMU = SINMU
00370              SINMU = -P
00371              D1 = (TWO*DMU-ONE)*(TWO*DMU+ONE)
00372              D2 = ZERO
00373              DIV = DDIV
00374              P = ZERO
00375              Q = ZERO
00376              Q0 = D1/DIV
00377              TERM = Q0
00378              DO 310 I = 2, 20
00379                  D2 = D2 + EIGHT

```

```

00380          D1 = D1 - D2
00381          DIV = DIV + DDIV
00382          TERM = -TERM*D1/DIV
00383          P = P + TERM
00384          D2 = D2 + EIGHT
00385          D1 = D1 - D2
00386          DIV = DIV + DDIV
00387          TERM = TERM*D1/DIV
00388          Q = Q + TERM
00389          IF (ABS(TERM) .LE. EPS) GO TO 320
00390          310      CONTINUE
00391          320      P = P + ONE
00392                  Q = Q + Q0
00393                  IF (K .EQ. 1) THEN
00394                      YA = SQ2BPI * (P*COSMU-Q*SINMU) / DEN
00395                      ELSE
00396                          YA1 = SQ2BPI * (P*COSMU-Q*SINMU) / DEN
00397                      END IF
00398                      DMU = DMU + ONE
00399          350      CONTINUE
00400                  END IF
00401                  IF (NA .EQ. 1) THEN
00402                      H = TWO*(ENU+ONE)/EX
00403                      IF (H .GT. ONE) THEN
00404                          IF (ABS(YA1) .GT. XINF/H) THEN
00405                              H = ZERO
00406                              YA = ZERO
00407                          END IF
00408                      END IF
00409                      H = H*YA1 - YA
00410                      YA = YA1
00411                      YA1 = H
00412                  END IF
00413          C-----
00414          C   Now have first one or two Y's
00415          C-----
00416                  BY(1) = YA
00417                  BY(2) = YA1
00418                  IF (YA1 .EQ. ZERO) THEN
00419                      NCALC = 1
00420                  ELSE
00421                      AYE = ONE + ALPHA
00422                      TWOBXYX = TWO/EX
00423                      NCALC = 2
00424                      DO 400 I = 3, NB
00425                          IF (TWOBXYX .LT. ONE) THEN
00426                              IF (ABS(BY(I-1))*TWOBXYX .GE. XINF/AYE)
00427                                  1      GO TO 450
00428                              ELSE
00429                                  IF (ABS(BY(I-1)) .GE. XINF/AYE/TWOBXYX )
00430                                      1      GO TO 450
00431                              END IF
00432                              BY(I) = TWOBXYX*AYE*BY(I-1) - BY(I-2)
00433                              AYE = AYE + ONE
00434                              NCALC = NCALC + 1
00435          400      CONTINUE
00436                  END IF
00437          450      DO 460 I = NCALC+1, NB
00438                      BY(I) = ZERO
00439          460      CONTINUE

```

```

00440         ELSE
00441             BY(1) = ZERO
00442             NCALC = MIN(NB,0) - 1
00443         END IF
00444     900 RETURN
00445 C----- Last line of RYBESL -----
00446     END

00001     REAL FUNCTION GAMMA(X)
00002 CD     DOUBLE PRECISION FUNCTION DGAMMA(X)
00003 C-----
00004 C
00005 C This routine calculates the GAMMA function for a real argument X.
00006 C Computation is based on an algorithm outlined in reference 1.
00007 C The program uses rational functions that approximate the GAMMA
00008 C function to at least 20 significant decimal digits. Coefficients
00009 C for the approximation over the interval (1,2) are unpublished.
00010 C Those for the approximation for X .GE. 12 are from reference 2.
00011 C The accuracy achieved depends on the arithmetic system, the
00012 C compiler, the intrinsic functions, and proper selection of the
00013 C machine-dependent constants.
00014 C
00015 C
00016 C*****
00017 C*****
00018 C
00019 C Explanation of machine-dependent constants
00020 C
00021 C beta    - radix for the floating-point representation
00022 C maxexp  - the smallest positive power of beta that overflows
00023 C XBIG    - the largest argument for which GAMMA(X) is representable
00024 C           in the machine, i.e., the solution to the equation
00025 C           GAMMA(XBIG) = beta**maxexp
00026 C XINF    - the largest machine representable floating-point number;
00027 C           approximately beta**maxexp
00028 C EPS     - the smallest positive floating-point number such that
00029 C           1.0+EPS .GT. 1.0
00030 C XMININ  - the smallest positive floating-point number such that
00031 C           1/XMININ is machine representable
00032 C
00033 C     Approximate values for some important machines are:
00034 C
00035 C           beta      maxexp      XBIG
00036 C
00037 C CRAY-1      (S.P.)      2          8191      966.961
00038 C Cyber 180/855
00039 C   under NOS  (S.P.)      2          1070     177.803
00040 C IEEE (IBM/XT,
00041 C   SUN, etc.) (S.P.)      2           128     35.040
00042 C IEEE (IBM/XT,
00043 C   SUN, etc.) (D.P.)      2          1024     171.624
00044 C IBM 3033    (D.P.)     16           63      57.574
00045 C VAX D-Format (D.P.)      2           127     34.844
00046 C VAX G-Format (D.P.)      2          1023     171.489
00047 C
00048 C           XINF      EPS      XMININ
00049 C
00050 C CRAY-1      (S.P.)    5.45E+2465  7.11E-15  1.84E-2466
00051 C Cyber 180/855
00052 C   under NOS  (S.P.)    1.26E+322  3.55E-15  3.14E-294

```

```

00053 C IEEE (IBM/XT,
00054 C   SUN, etc.)   (S.P.)   3.40E+38   1.19E-7   1.18E-38
00055 C IEEE (IBM/XT,
00056 C   SUN, etc.)   (D.P.)   1.79D+308   2.22D-16   2.23D-308
00057 C IBM 3033      (D.P.)   7.23D+75   2.22D-16   1.39D-76
00058 C VAX D-Format  (D.P.)   1.70D+38   1.39D-17   5.88D-39
00059 C VAX G-Format  (D.P.)   8.98D+307   1.11D-16   1.12D-308
00060 C
00061 C*****
00062 C*****
00063 C
00064 C Error returns
00065 C
00066 C The program returns the value XINF for singularities or
00067 C   when overflow would occur. The computation is believed
00068 C   to be free of underflow and overflow.
00069 C
00070 C
00071 C Intrinsic functions required are:
00072 C
00073 C   INT, DBLE, EXP, LOG, REAL, SIN
00074 C
00075 C
00076 C References: "An Overview of Software Development for Special
00077 C   Functions," W. J. Cody, Lecture Notes in Mathematics,
00078 C   506, Numerical Analysis Dundee, 1975, G. A. Watson
00079 C   (ed.), Springer Verlag, Berlin, 1976.
00080 C
00081 C   Computer Approximations, Hart, Et. Al., Wiley and
00082 C   sons, New York, 1968.
00083 C
00084 C Latest modification: October 12, 1989
00085 C
00086 C Authors: W. J. Cody and L. Stoltz
00087 C   Applied Mathematics Division
00088 C   Argonne National Laboratory
00089 C   Argonne, IL 60439
00090 C
00091 C-----
00092 C   INTEGER I,N
00093 C   LOGICAL PARITY
00094 C   REAL
00095 CD   DOUBLE PRECISION
00096 C   1   C,CONV,EPS,FACT,HALF,ONE,P,PI,Q,RES,SQRTPI,SUM,TWELVE,
00097 C   2   TWO,X,XBIG,XDEN,XINF,XMININ,XNUM,Y,Y1,YSQ,Z,ZERO
00098 C   DIMENSION C(7),P(8),Q(8)
00099 C-----
00100 C Mathematical constants
00101 C-----
00102 C   DATA ONE,HALF,TWELVE,TWO,ZERO/1.0E0,0.5E0,12.0E0,2.0E0,0.0E0/,
00103 C   1   SQRTPI/0.9189385332046727417803297E0/,
00104 C   2   PI/3.1415926535897932384626434E0/
00105 CD   DATA ONE,HALF,TWELVE,TWO,ZERO/1.0D0,0.5D0,12.0D0,2.0D0,0.0D0/,
00106 CD   1   SQRTPI/0.9189385332046727417803297D0/,
00107 CD   2   PI/3.1415926535897932384626434D0/
00108 C-----
00109 C Machine dependent parameters
00110 C-----
00111 C   DATA XBIG,XMININ,EPS/35.040E0,1.18E-38,1.19E-7/,
00112 C   1   XINF/1.7E38/

```

```

00113 CD DATA XBIG,XMININ,EPS/171.624D0,2.23D-308,2.22D-16/,
00114 CD 1 XINF/1.79D308/
00115 C-----
00116 C Numerator and denominator coefficients for rational minimax
00117 C approximation over (1,2).
00118 C-----
00120 DATA P/-1.71618513886549492533811E+0,2.47656508055759199108314E+1,
00121 1 -3.79804256470945635097577E+2,6.29331155312818442661052E+2,
00122 2 8.66966202790413211295064E+2,-3.14512729688483675254357E+4,
00123 3 -3.61444134186911729807069E+4,6.64561438202405440627855E+4/
00124 DATA Q/-3.08402300119738975254353E+1,3.15350626979604161529144E+2,
00125 1 -1.01515636749021914166146E+3,-3.10777167157231109440444E+3,
00126 2 2.25381184209801510330112E+4,4.75584627752788110767815E+3,
00127 3 -1.34659959864969306392456E+5,-1.15132259675553483497211E+5/
00128 CD DATA P/-1.71618513886549492533811D+0,2.47656508055759199108314D+1,
00129 CD 1 -3.79804256470945635097577D+2,6.29331155312818442661052D+2,
00130 CD 2 8.66966202790413211295064D+2,-3.14512729688483675254357D+4,
00131 CD 3 -3.61444134186911729807069D+4,6.64561438202405440627855D+4/
00132 CD DATA Q/-3.08402300119738975254353D+1,3.15350626979604161529144D+2,
00133 CD 1 -1.01515636749021914166146D+3,-3.10777167157231109440444D+3,
00134 CD 2 2.25381184209801510330112D+4,4.75584627752788110767815D+3,
00135 CD 3 -1.34659959864969306392456D+5,-1.15132259675553483497211D+5/
00136 C-----
00137 C Coefficients for minimax approximation over (12, INF).
00138 C-----
00139 DATA C/-1.910444077728E-03,8.4171387781295E-04,
00140 1 -5.952379913043012E-04,7.93650793500350248E-04,
00141 2 -2.777777777777681622553E-03,8.3333333333333331554247E-02,
00142 3 5.7083835261E-03/
00143 CD DATA C/-1.910444077728D-03,8.4171387781295D-04,
00144 CD 1 -5.952379913043012D-04,7.93650793500350248D-04,
00145 CD 2 -2.777777777777681622553D-03,8.3333333333333331554247D-02,
00146 CD 3 5.7083835261D-03/
00147 C-----
00148 C Statement functions for conversion between integer and float
00149 C-----
00150 CONV(I) = REAL(I)
00151 CD CONV(I) = DBLE(I)
00152 PARITY = .FALSE.
00153 FACT = ONE
00154 N = 0
00155 Y = X
00156 IF (Y .LE. ZERO) THEN
00157 C-----
00158 C Argument is negative
00159 C-----
00160 Y = -X
00161 Y1 = AINT(Y)
00162 RES = Y - Y1
00163 IF (RES .NE. ZERO) THEN
00164 IF (Y1 .NE. AINT(Y1*HALF)*TWO) PARITY = .TRUE.
00165 FACT = -PI / SIN(PI*RES)
00166 Y = Y + ONE
00167 ELSE
00168 RES = XINF
00169 GO TO 900
00170 END IF
00171 C-----
00172 C Argument is positive

```

```

00173 C-----
00174         IF (Y .LT. EPS) THEN
00175 C-----
00176 C   Argument .LT. EPS
00177 C-----
00178         IF (Y .GE. XMININ) THEN
00179             RES = ONE / Y
00180         ELSE
00181             RES = XINF
00182             GO TO 900
00183         END IF
00184     ELSE IF (Y .LT. TWELVE) THEN
00185         Y1 = Y
00186         IF (Y .LT. ONE) THEN
00187 C-----
00188 C   0.0 .LT. argument .LT. 1.0
00189 C-----
00190             Z = Y
00191             Y = Y + ONE
00192         ELSE
00193 C-----
00194 C   1.0 .LT. argument .LT. 12.0, reduce argument if necessary
00195 C-----
00196             N = INT(Y) - 1
00197             Y = Y - CONV(N)
00198             Z = Y - ONE
00199         END IF
00200 C-----
00201 C   Evaluate approximation for 1.0 .LT. argument .LT. 2.0
00202 C-----
00203         XNUM = ZERO
00204         XDEN = ONE
00205         DO 260 I = 1, 8
00206             XNUM = (XNUM + P(I)) * Z
00207             XDEN = XDEN * Z + Q(I)
00208 260     CONTINUE
00209         RES = XNUM / XDEN + ONE
00210         IF (Y1 .LT. Y) THEN
00211 C-----
00212 C   Adjust result for case 0.0 .LT. argument .LT. 1.0
00213 C-----
00214             RES = RES / Y1
00215             ELSE IF (Y1 .GT. Y) THEN
00216 C-----
00217 C   Adjust result for case 2.0 .LT. argument .LT. 12.0
00218 C-----
00219             DO 290 I = 1, N
00220                 RES = RES * Y
00221                 Y = Y + ONE
00222 290     CONTINUE
00223             END IF
00224         ELSE
00225 C-----
00226 C   Evaluate for argument .GE. 12.0,
00227 C-----
00228             IF (Y .LE. XBIG) THEN
00229                 YSQ = Y * Y
00230                 SUM = C(7)
00231                 DO 350 I = 1, 6
00232                     SUM = SUM / YSQ + C(I)

```

```

00233      350          CONTINUE
00234          SUM = SUM/Y - Y + SQRTPI
00235          SUM = SUM + (Y-HALF)*LOG(Y)
00236          RES = EXP(SUM)
00237      ELSE
00238          RES = XINF
00239          GO TO 900
00240      END IF
00241  END IF
00242  C-----
00243  C   Final adjustments and return
00244  C-----
00245      IF (PARITY) RES = -RES
00246      IF (FACT .NE. ONE) RES = FACT / RES
00247      900 GAMMA = RES
00248  CD900 DGAMMA = RES
00249      RETURN
00250  C ----- Last line of GAMMA -----
00251      END

00001  C   **** INLET FAN RADIATION SUBROUTINE, E. J. RICE, REVISED 02/21/1998
00002  C   ** CONTAINS PLANE WAVE RADIATION AND IMPROVED TERMINATION LOSS
00003  C   ** THE FOLLOWING "BBRDCFIN" IS THE MAIN SUBROUTINE. ALL ARGUMENTS
00004  C   ARE INPUTS EXCEPT "NANGLE," "ANGLE," "SPL," "SPLTL" AND WATTS
00005  C   RETURNED TO THE MAIN PROGRAM FOR PRINTING, FILE STORAGE ETC.
00006  C
00007  C   ** THIS VERSION REMOVES THE P**2 OSCILLATIONS BEYOND THE PEAK. FOUR
00008  C   REGIONS (ETA AND CUT-OFF RATIO) DETERMINE PROPER APPROXIMATION FOR
00009  C   HIGH ANGLE RADIATION. DEVELOPED 01/30/98.
00010
00011      SUBROUTINE BBRDCFIN(TTOT,PTOT,DISTANCE,
00012      1ISIDELN,DIAM,ALIP,BLIP,FMACHI,FMACHS,NCOF,WATTSCOF,ETAI,
00013      2DELANG,NANGLE,ANGLE,SPL,SPLTL,WATTS,WATTRAN)
00014  C
00015      DIMENSION ANGLE(200),SPL(200),SPLTL(200),WATTSCOF(200),
00016      1COFRAT(200),PSQTOT(200),PSQTLOS(200),PSQRADT(200)
00017
00018      COMMON FMSQ,FM1,BETA,COFBETIN,CFBTINSQ,GDEN,HDEN,
00019      1PSQPK,PSIC,AC,BC,CC,IREG
00020
00021  C
00022  C   ** SUBROUTINES REQUIRED "LIPEF3" AND "PSQGCOF"
00023  C
00024  C   ***** DEFINITION OF SUBROUTINE ARGUMENTS *****
00025  C
00026  C   TTOT   ABSOLUTE TEMPERATURE, (DEGREES RANKINE)
00027  C   PTOT   ABSOLUTE PRESSURE, (PSIA)
00028  C   DISTANCE  RADIUS OR SIDELINE DISTANCE OF MICROPHONE ARRAY, (FT.)
00029  C   ISIDELN = 0 FOR CONSTANT RADIUS, = 1 FOR CONSTANT SIDELINE CALCULATION
00030  C   DIAM    INLET DUCT DIAMETER, (INCHES)
00031  C   ALIP    MAJOR OR AXIAL RUNNING DIMENSION OF ELLIPTIC INLET LIP (INCHES)
00032  C   BLIP    MINOR OR TRANSVERSE DIMENSION OF ELLIPTIC INLET LIP (INCHES)
00033  C   FMACH   INLET MACH NUMBER, NEGATIVE FOR INLET
00034  C   FMACHS  SURROUNDING MACH NUMBER, ALSO NEGATIVE FOR INLET
00035  C   NCOF    NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
00036  C   WATTSCOF VECTOR WITH ACOUSTIC POWER FOR EACH BIN, (WATTS)
00037  C   ETA     FREQUENCY PARAMETER, (DUCT DIAMETER)/(SOUND WAVELENGTH)
00038  C   DELANG  ANGLE INCREMENT DESIRED ON FAR-FIELD RADIAL TRAVERSE, (DEGREES)
00039  C   NANGLE  NUMBER OF ANGLES USED IN CALCULATION, DETERMINED BY DELANG AND

```

```

      C          THE MAXIMUM ANGLE OF 90 DEGREES
00041  C ANGLE    VECTOR OF ANGLES ON THE FAR-FIELD TRAVERSE, (DEGREES)
00042  C SPL      THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
00043  C          "IGNORING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
00044  C          2*10**(-5) NEWTONS/METER**2
00045  C SPLTL    THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
      C          "INCLUDING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
      C          2*10**(-5) NEWTONS/METER**2
00048  C WATTS    SUM OF ACOUSTIC POWER IN ALL THE CUT-OFF RATIO BINS, (WATTS)
00049  C WATTRAN  SUM OF TRANSMITTED ACOUSTIC POWER, ALL BINS, (WATTS)
00050
00051          FMACH = FMACHI
00052          ETA = ETAI
00053          DRAD = 0.5*DIAM
00054          PI = 3.1415927
00055          AREAD = PI*DRAD**2
00056          ABELEX = AREAD+2.0*PI*BLIP*(BLIP+0.5*PI*DRAD)
00057          FMBELEX = FMACH*AREAD/ABELEX
00058          ETABELEX = ETA*(DIAM+2.0*BLIP)/DIAM
00059
      C  CALCULATE COEF. (PLANE WAVE) INTEGRATION OF RAD. DIRECTIVITY, MACH=0
      IF(ETABELEX.LT.1.0) THEN
00062          ACOEFPW = 0.741697+3.190822*ETABELEX**2.650078
00063          GO TO 14
00064          END IF
00065          ACOEFPW = 3.932518*ETABELEX**1.96285
00066      14 CONTINUE
00067  C
00068
00069          PSQCOEFP = ACOEFPW
00070
00071
00072          QM = 1.0+0.2*FMACHS**2
00073          TSUR = TTOT/QM
00074          PSUR = PTOT/QM**3.5
00075
00076          SONIC = 49.0422*SQRT(TSUR)
00077          RHO = 144.*PSUR/(53.3*TSUR)
00078          POWCON = 8.36424*RHO*SONIC
00079
00080          ETAEXP = ETA**1.08156
00081          WATINFIX = (1.0+1.9036*ETAEXP)/(PI*0.71385*ETAEXP)
00082
00083  C ***** DIMENSIONS, SONIC (FT/SEC), RHO (LBm/FT**3) *****
00084
00085  C ***** NOTE!! THIS VERSION CALCULATES TO 178 DEGREES FROM INLET AXIS
00086
00087          NANGLE = 178.0/DELANG
00088          DO 5 I=1,NANGLE
00089              FI = I
00090              ANGLE(I) = FI*DELANG
00091      5 CONTINUE
00092
00093          FMSQ = FMACH**2
00094          FM1 = 1.0-FMSQ
00095          BETA = SQRT(FM1)
00096
00097          ACOEF = 0.7/ETA
00098
00099          FCOF = NCOF

```

```

00100      FCOFINV = 1./FCOF
00101      FCOFIND2 = 0.5/FCOF
00102  C ***** SET UP CUT-OFF RATIOS IN THE DUCT *****
00103      COFSQPR = 1.0
00104      DO 20 I=1,NCOF
00105          COFRAT(I) = SQRT(1.0/(COFSQPR-FCOFIND2))
00106      COFSQPR = COFSQPR-FCOFINV
00107      20 CONTINUE
00108
00109  C ***** INITIALIZE P**2 AT EACH FAR-FIELD ANGLE *****
00110  C
00111      DO 10 I=1,NANGLE
00112          PSQRADT(I) = 0.0
00113          PSQTLOS(I) = 0.0
00114          PSQTOT(I) = 0.0
00115      10 CONTINUE
00116  C
00117  C ***** START LOOP ON CUT-OFF RATIO *****
00118  C
00119      WATTS = 0.0
00120      WATTRAN = 0.0
00121
00122      DO 70 J=1,NCOF
00123          WATTS = WATTS+WATTSCOF(J)
00124          POWCOEF = POWCON*WATTSCOF(J)
00125
00126  C *****
00127  C !!!!!!! VERY IMPORTANT! FOLLOWING DEFINES PLANE WAVE REGIME !!!!!
00128  C *****
00129
00130  C ***** DETERMINE IF PLANE WAVE RADIATION ALGORITHM SHOULD BE USED
00131  C ***** IF PLANE WAVE IPW = 1 (YES) FOR COFRATD>=3 OR LAST BIN
00132      IPW = 0
00133      IF(COFRAT(J).GE.3.0.OR.J.EQ.NCOF) IPW=1
00134  C
00135  C ***** PROGRAM SPITS HERE. IT WILL CONTINUE ON FOR NON-PLANE
00136  C ***** WAVES AND WILL JUMP FOR THE PLANE WAVE
00137
00138      IF(IPW.EQ.1) GO TO 45
00139
00140  C ***** NOW IN "NON-PLANE" WAVE, RADIAL OR SPINNING MODE REGIME !!
00141
00142
00143
00144      XID = COFRAT(J)
00145      FRAC = 0.85
00146      IF(XID.LE.2.5) FRAC=1.0-0.1*(XID-1.0)
00147  C
00148  C ** FOLLOWING ALLOWS SHARP EDGE OR UNFLANGED DUCT APPROXIMATION
00149      BTHK = BLIP/DIAM
00150      IF(BTHK.LT.0.01.OR.XID.GT.10.0) THEN
00151          XIBEL = XID
00152          FMBEL = FMACH
00153          DIAMBEL = DIAM
00154          XDARAD = 1.0
00155          GO TO 23
00156      END IF
00157  C
00158      CALL LIPEF3(XID,XIBEL,FMACH,FMBEL,DIAMBEL,DIAM,ALIP,BLIP,XDARAD,FR
00159      1AC)

```

```

00160      IF(FMACH.NE.0.0) THEN
00161      FMBEL = FMACH*ABS(FMBEL/FMACH)
00162      GO TO 23
00163      END IF
00164      FMBEL = 0.0
00165      C
00166      C
00167      23 CONTINUE
00168      ETABEL = ETA*DIAMBEL/DIAM
00169      FMSQ = FMBEL**2
00170      FM1 = 1.0-FMSQ
00171      BETA = SQRT(FM1)
00172
00173      COF = XIBEL
00174      COFINV = 1.0/XIBEL
00175      COFINVSQ = COFINV**2
00176      COFBETIN = 1.0/(XIBEL*BETA)
00177      CFBTINSQ = COFBETIN**2
00178      COFM1 = 1.0-COFINVSQ
00179      COFSQRT = SQRT(COFM1)
00180
00181      A90 = 2.0*(ACOE+COFSQRT)/(ACOE+1.0)
00182      PSQCOEF = A90*(1.0-FMSQ*COFM1)**1.5/BETA
00183      PSQCOEF = PSQCOEF*WATINFIX
00184
00185      GDEN = (1.0+COFSQRT)**2
00186      COSPK = BETA*COFSQRT/SQRT(1.0-FMSQ*COFM1)
00187      PSIPK = ACOS(COSPK)*180.0/PI
00188      HDEN = 1.0-FMBEL*COSPK
00189
00190
00191      C ***** CALCULATE TRANSMISSION LOSS IN NON-PLANE WAVE REGION *****
00192
00193      QF = PI*ETABEL*(1.0-1.0/XIBEL)
00194      QF15SQ = (QF-1.5)**2
00195      RADREC = 1.135*EXP(-0.29*(QF+0.18)**2)
00196
00197      IF(QF.LE.1.5) THEN
00198      RADRES = 1.5*EXP(-0.2124*QF15SQ)
00199      GO TO 53
00200      END IF
00201
00202      RADRES = 1.0+0.5*EXP(-0.5338*QF15SQ)
00203      53 CONTINUE
00204
00205      TAU = SQRT(1.0-1.0/XIBEL**2)
00206      TPM = TAU+FMBEL
00207      TTM = TAU*FMBEL
00208      QDEN = (RADRES*TPM+TTM+1.0)**2+(RADREC*TPM)**2
00209      QNUM = (RADRES+FMBEL)*(RADRES*FMBEL+1.0)+FMBEL*RADREC**2
00210      TLCF = 4.0*TAU*QNUM/QDEN
00211
00212      IF(TLCF.GT.1.0) TLCF=1.0
00213      IF(TLCF.LT.0.0) TLCF=0.0001
00214
00215      C ***** FINISHED WITH TRANSMISSION LOSS AT CURRENT CUT-OFF RATIO **
00216
00217      C ***** CALCULATE TRANSMITTED POWER *****
00218      C
00219      WATTRAN = WATTRAN+TLCF*WATTSCOF(J)

```

```

00220 C
00221 C ***** CALCULATE P**2 COEF. TO ACCOUNT FOR WATTS.
00222 C TO ACCOUNT FOR MIC. RADIUS, DIVIDE BY DISTANCE**2 AT PSQ BELOW.
00223 C
00224
00225 PSQPK = PSQCOEF*ETABEL*XIBEL/(2.0*BETA)
00226 C
00227 C ***** START SORTING INTO REGIMES TO HANDLE LARGE ANGLES *****
00228 C
00229 IREG = 0
00230 ETAC1 = 0.6*BETA/(1.0-COFINV)
00231 IF(ETABEL.GT.ETAC1) THEN
00232 IREG = 1
00233
00234 C ***** REGION 1, SLOPE AT ARG=PI/2 USED TO PROJECT BEYOND PEAK
00235
00236 EPS = 1.0/(BETA*COF)+0.5/ETABEL
00237 EPSQ = EPS**2
00238 QNUM = 1.0+FMSQ*EPSQ
00239 DEPDPSI = QNUM*SQRT(1.0-FM1*EPSQ)
00240 PSQRATC1 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00241 FDEN = CFBTINSQ-EPSQ
00242 DPSQDPSI = (FDEN+4.0*EPSQ)*PSQRATC1*DEPDPSI/(EPS*FDEN)
00243 SINPSIC1 = EPS/SQRT(QNUM)
00244 PSIC = ASIN(SINPSIC1)*180.0/PI
00245 AC = ALOG(PSQRATC1)
00246 BC = 0.8889*DPSQDPSI/PSQRATC1
00247 BC = BC*PI/180.0
00248 CC = -0.1781*BC
00249 GO TO 50
00250 END IF
00251
00252 ETAC2 = 0.6*BETA*COF
00253 IF(ETABEL.GT.ETAC2) THEN
00254 IREG = 2
00255
00256 C ***** REGION 2, PSQratio AT ARG=-PI/2 USED TO FIT EXPONENTIAL FOR
00257 C USE BEYOND PEAK
00258
00259 EPS = 1.0/(BETA*COF)-0.5/ETABEL
00260 PSQRATC2 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00261 EPSQ = EPS**2
00262 QDEN = 1.0+FMSQ*EPSQ
00263 SINPSIC2 = EPS/SQRT(QDEN)
00264 PSIC2 = ASIN(SINPSIC2)*180.0/PI
00265 AC = ALOG(PSQRATC2)/(PSIPK-PSIC2)**2
00266 GO TO 50
00267 END IF
00268
00269
00270
00271 C ***** REGION 3, LOW ETA REGION, PSIPK>60 DEG. FIT EXPONENTIAL AT
C 0.5*PSIPK FOR USE BEYOND PEAK. CODE IREG=3. EXPONENTIAL EQUATION
C USED IN PSQ SUBROUTINE FOR PSI > PSIPK.
00274
00275 IF(PSIPK.GT.60.0) THEN
00276
00277 IREG = 3
00278
00279 ANGF = 0.5*PSIPK

```

```

00280      ANGRAD = ANGF*PI/180.0
00281      SIN F = SIN(ANGRAD)
00282      EPS = SIN F/SQRT(1.0-FMSQ*SIN F**2)
00283      ARG = PI*ETABEL*(COFBETIN-EPS)
00284      SINARG = SIN(ARG)
00285      PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00286      PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00287      AC = ALOG(PSQRATC3)/(PSIPK-ANGF)**2
00288      GO TO 50
00289      END IF
00290
00291 C ***** REGION 4, LOW ETA REGION, PSIPK<60 DEG. FIT EXPONENTIAL AT
00292 C      80 deg. FOR USE BEYOND PEAK. CODE IREG=4. EXPONENTIAL EQUATION
00293 C      USED IN PSQ SUBROUTINE FOR PSI>80 deg.
00294
00295      IREG = 4
00296
00297      ANGF = 80.0
00298      ANGRAD = ANGF*PI/180.0
00299      SIN F = SIN(ANGRAD)
00300      EPS = SIN F/SQRT(1.0-FMSQ*SIN F**2)
00301      ARG = PI*ETABEL*(COFBETIN-EPS)
00302      SINARG = SIN(ARG)
00303      PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00304      PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00305      AC = ALOG(PSQRATC3)/(PSIPK-ANGF)**2
00306
00307 50 CONTINUE
00308
00309 C
00310      DO 25 I=1,NANGLE
00311      ANG = ANGLE(I)
00312      IF(ANG.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00313      PSQRADT(I) = 0.0
00314      PSQTOT(I) = 0.0
00315      PSQTL0S(I) = 0.0
00316      GO TO 25
00317      END IF
00318 C
00319      CALL PSQGC0F(ANG,PSQ,FMBEL,ETABEL,XIBEL,PSIPK)
00320
00321      PSQRAD = PSQ
00322      RAD = DISTANCE
00323      IF(ISIDELN.EQ.1) THEN
00324      RAD = DISTANCE/SIN(ANG*PI/180.0)
00325      END IF
00326      PSQ = PSQ/RAD**2
00327
00328      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00329      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00330      PSQTL0S(I) = PSQTL0S(I)+POWCOEF*PSQ*TL0CF
00331 C ***** NOTE THAT AN INLET TRANSMISSION LOSS (TL0CF) HAS BEEN USED *****
00332 25 CONTINUE
00333
00334      GO TO 70
00335
00336
00337 45 CONTINUE
00338 C ***** IN PLANE-WAVE CALCULATION PROCEDURE !!!!!!!!!!!!!!!!!!!!!!!
00339

```

```

00340
00341 C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
00342 PSQPK = 2.0*PSQCOEFP
00343 C *****
00344
00345 GDEN = 4.0
00346
00347 FMSQEX = FMBELEX**2
00348
00349 C ***** LOW ETA REGION. FIT EXPONENTIAL AT PSICRPL FOR BEYOND PEAK.
00350 C ***** IF PSICRPL < 90 CAN'T BE OBTAINED, USE PSICRPL = 90.
00351 ANGF = 90.0
00352 ETACRPL = 0.5*SQRT(1.0-FMSQEX)
00353 SINCRPL = 1.0/SQRT(4.0*ETABELEX**2+FMSQEX)
00354 IF(ETABELEX.GT.ETACRPL) ANGF=ASIN(SINCRPL)*180./PI
00355 PSICRPL = ANGF
00356 ANGFRAD = ANGF*PI/180.0
00357 SINF = SIN(ANGFRAD)
00358 ARG = PI*ETABELEX*SINF/SQRT(1.0-FMSQEX*SINF**2)
00359 SINARG = SIN(ARG)
00360 PSQRATPL = (SINARG/ARG)**2
00361 ACPL = ALOG(PSQRATPL)/ANGF**2
00362
00363 C ***** ACPL IS THE AMPLITUDE OF THE EXPONENTIAL AT PSICRPL *****
00364 C ***** END LOW ETA REGION EXPONENTIAL FIT AT PSICRPL. *****
00365
00366 C ** CALCULATE TRANSMISSION LOSS AT BELMOUTH EXIT CUT-OFF RATIO AND
00367 C ** FREQUENCY PARAMETER.
00368
00369 X = 0.5*(PI*ETABELEX)**2
00370 RADRES = 1.0+X*EXP(-0.325226*X)-
00371 1 EXP(-0.101669*ETABELEX**5.7848)
00372 A = 0.023567
00373 Y = 0.5*PI**2*ETABELEX
00374 RADREC=EXP(-3.574331*ETABELEX**1.957292)*8.*ETABELEX/
00375 13.+A*Y**2/(1.+A*Y**3)
00376 QDEN = (1.0+FMBELEX)**2*((RADRES+1.0)**2+RADREC**2)
00377 TLCF=4.*(RADRES*(1.+FMSQEX)+FMBELEX*(RADRES**2+
00378 1 RADREC**2+1.))/QDEN
00379 IF(TLCF.GT.1.0) TLCF=1.0
00380 IF(TLCF.LT.0.0) TLCF=0.0001
00381
00382 C ** END TRANSMISSION LOSS CALCULATION FOR PLANE WAVE AT BELLMOUTH EXIT
00383
00384 C ***** CALCULATE TRANSMITTED POWER *****
00385 C
00386 WATTRAN = WATTRAN+TLCF*WATTSCOF(J)
00387 C
00388
00389 DO 40 I=1,NANGLE
00390 FI = I
00391 ANGDEG = ANGLE(I)
00392 IF(ANGDEG.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00393 PSQRADT(I) = 0.0
00394 PSQTOT(I) = 0.0
00395 PSQTLOS(I) = 0.0
00396 GO TO 40
00397 END IF
00398 C
00399 ANGRAD = ANGDEG*PI/180.0
00400

```

```

00401      SINANG = SIN(ANGRAD)
00402      COSANG = COS(ANGRAD)
00403
00404      Q1DEN = SQRT(1.0-FMSQEX*SINANG**2)
00405      Q1 = SINANG/Q1DEN
00406      ARG = PI*ETABELEX*Q1
00407      SINSQNUM = (SIN(ARG))**2
00408      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00409      PSQRAT = 1.0
00410
00411      PSQDEN = ARG**2
00412      IF(PSQDEN.LT.1.E-06.AND.ANGDEG.LE.90.0) GO TO 49
00413
00414
00415      C ** LOW FREQUENCY REGION. FOLLOWING ALLOWS REASONABLE
      C ** EXTRAPOLATION BEYOND 90 deg, STARTING AT ARG = PI/2 IF ETA>ETAcrit
      C ** OR AT 90 deg. OTHERWISE. USES EXPONENTIAL FIT AT PSI = 0
00418      C ** AND PSI = PSIFIT.
00419
00420      IF(ANGDEG.LT.PSICRPL) GO TO 48
00421
00422      QEXP = ACPL*(ANGDEG)**2
00423      IF(QEXP.LT.-20.) QEXP=-20.
00424
00425      PSQRAT = EXP(QEXP)
00426      GO TO 49
00427
00428
00429      48 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00430      49 CONTINUE
00431
00432      PSQ = PSQRAT*PSQPK*GG
00433
00434      PSQRAD = PSQ
00435      C
00436      RAD = DISTANCE
00437      IF(ISIDELN.EQ.1) THEN
00438      RAD = DISTANCE/SIN(ANGDEG*PI/180.0)
00439      END IF
00440
00441      PSQ = PSQ/RAD**2
00442
00443      C
00444      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00445      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00446      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF
00447
00448      C ***** NOTE THAT A TRANSMISSION LOSS (TLCF) HAS BEEN USED *****
00449
00450      40 CONTINUE
00451      41 CONTINUE
00452
00453
00454      70 CONTINUE
00455
00456      FNANGLE = NANGLE
00457      SUMWATT = 0.0
00458      DO 75 I=1,NANGLE
00459      ANGRAD = ANGLE(I)*PI/180.0
00460      SUMWATT = SUMWATT+PSQRADT(I)*SIN(ANGRAD)

```

```

00461
00462     IF(PSQTOT(I).LT.4.E-08) THEN
00463     SPLTL(I) = 20.0
00464     SPL(I) = 20.0
00465     GO TO 75
00466     END IF
00467     SPLTL(I) = 10.0*ALOG10(PSQTLOS(I))+93.9794
00468     SPL(I) = 10.0*ALOG10(PSQTOT(I))+93.9794
00469 75 CONTINUE
00470
00471     WATTINT = SUMWATT*PI/(POWCON*FNANGLE)
00472     SPLDIF = 10.0*ALOG10(WATTS/WATTINT)
00473
00474     DO 80 I=1,NANGLE
00475     SPLTL(I) = SPLTL(I)+SPLDIF
00476     SPL(I) = SPL(I)+SPLDIF
00477 80 CONTINUE
00478
00479
00480     RETURN
00481     END

00001 C
00002 C *****
00003 C ***** END OF MAIN SUBROUTINE "BBRDCFIN" *****
00004 C ***** MODIFIED 02/21/1998, E. J. RICE *****
00005 C *****
00006 C
00007 C *****
00008 C ** SUBROUTINE FOR CALC PSQ FOR EQUAL ENERGY PER MODE AT AN ANGLE
00009 C ** CUT-OFF RATIO APPROXIMATE EQUATIONS USED, BLOCK BUILD-UP AS IN
00010 C AIAA PAPER 96-1774, EMPIRICAL NORMALIZATION REPLACES FACTOR
00011 C Sqrt(1.-1/XI**2). FOUR REGIONS (ETA AND CUT-OFF RATIO) DETERMINE
00012 C PROPER APPROXIMATION FOR HIGH ANGLE RADIATION. DEVELOPED 01/30/98.
00013 C
00014 SUBROUTINE PSQGCOF(ANG,PSQ,FMACH,ETA,XI,PSIPK)
00015 C
00016 COMMON FMSQ,FM1,BETA,COFBETIN,CFBTINSQ,GDEN,HDEN,
00017 1PSQPK,PSIC,AC,BC,CC,IREG
00018
00019 PI = 3.1415927
00020
00021 ANGRAD = ANG*PI/180.0
00022 SINANG = SIN(ANGRAD)
00023 COSANG = COS(ANGRAD)
00024
00025 Q1DEN = Sqrt(1.0-FMSQ*SINANG**2)
00026 Q1 = SINANG/Q1DEN
00027 ARG = PI*ETA*(Q1-COFBETIN)
00028 SINSQNUM = (SIN(ARG))**2
00029 GG = (1.0+COSANG/Q1DEN)**2/GDEN
00030 HH = (1.0-FMACH*COSANG)/HDEN
00031 PSQRAT = 4.0*Q1/(BETA*XI*(Q1+COFBETIN)**2)
00032
00033 PSQDEN = ARG**2
00034 ANGCK = PSIPK+1.0
00035
00036 IF(ANG.GT.ANGCK) GO TO 5
00037 C ***** CHECK FOR 0/0, USE LIMITING CASE IF INDETERMINATE *****

```

```

00038 C ***** DON'T CHECK IF ANGLE ABOVE PEAK ANGLE *****
00039     IF(PSQDEN.LT.1.E-06) GO TO 39
00040     5 CONTINUE
00041
00042     IF(ANG.LT.PSIPK) GO TO 38
00043
00044     IF(ANG.GE.PSIC.AND.IREG.EQ.1) THEN
00045
00046 C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00047 C ** USES SLOPE AT ARG = PI/2 TO EXTRAPOLATE TO HIGH ANGLES *****
00048     DANG = ANG-PSIC
00049     QEXP = AC+BC*DANG/(1.0+CC*DANG)
00050     IF(QEXP.LT.-20.) QEXP=-20.
00051
00052     PSQRAT = EXP(QEXP)
00053     GO TO 39
00054     END IF
00055
00056     IF(ANG.GE.PSIPK.AND.IREG.EQ.2) THEN
00057
00058 C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00059 C ** USES EXPONENTIAL FIT AT ARG = -PI/2 TO EXTRAPOLATE TO HIGH ANGLES
00060 C ** ALSO HANDLES LOW FREQUENCY REGIMES 3 AND 4 FOR ANGLES BEYOND PEAK.
00061
00062     QEXP = AC*(ANG-PSIPK)**2
00063     IF(QEXP.LT.-20.) QEXP=-20.
00064
00065     PSQRAT = EXP(QEXP)
00066     GO TO 39
00067     END IF
00068
00069     IF(ANG.GE.PSIPK.AND.IREG.EQ.3) THEN
00070
00071 C ** ONE OF THE LOW FREQUENCY REGIONS, REGION 3. FOLLOWING ALLOWS
00072 C ** REASONABLE EXTRAPOLATION BEYOND 90 deg, STARTING AT PEAK.
00073 C ** USES EXPONENTIAL FIT AT PSI = PSIPK AND 0.5*PSIPK. PSIPK>60 deg.
00074
00075     QEXP = AC*(ANG-PSIPK)**2
00076     IF(QEXP.LT.-20.) QEXP=-20.
00077
00078     PSQRAT = EXP(QEXP)
00079     GO TO 39
00080     END IF
00081
00082 C ** ONLY REGION LEFT, REGION 4, WITH PSIPK<60 deg.
00083 C ** ONE OF THE LOW FREQUENCY REGIONS. FOLLOWING ALLOWS REASONABLE
00084 C ** EXTRAPOLATION BEYOND 90 deg, STARTING THIS TIME AT 80 deg.
00085 C ** USES EXPONENTIAL FIT AT PSI = PSIPK AND 80 deg. PSIPK<60 deg.
00086
00087     IF(ANG.LT.80.0) GO TO 38
00088
00089     QEXP = AC*(ANG-PSIPK)**2
00090     IF(QEXP.LT.-20.) QEXP=-20.
00091
00092     PSQRAT = EXP(QEXP)
00093     GO TO 39
00094
00095     38 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00096     39 CONTINUE
00097
00098     PSQ = PSQRAT*PSQPK*GG*HH

```

```

00099      C
00100
00101      RETURN
00102      END

00001      C*****
00002      C  ** END OF PSQCOF.FOR *****
00003      C  *****
00004
00005
00006
00007
00008      C
00009      C  *****
00010      C  *****  ALLOWS CUT OFF MODES TO PROPAGATE IN BELLMOUTH IF THEY CAN
00011      C  *****  BE CUT ON OR PROPAGATING BEFORE THE BELLMOUTH EXIT *****
00012      C  *****
00013      C  ** SUBROUTINE "LIPEF3" CALCULATES THE EFFECT OF THE INLET LIP OR
00014      C  BELLMOUTH ON THE INLET FAR-FIELD RADIATION, SECOND MODEL
00015      C  *****
00016      C  *****
00017      C
00018      SUBROUTINE LIPEF3(XID,XIBEL,FMACK,FMBEL,DBEL,DIAM,ALIP,BLIP,XDARAD
00019      1,FRAC)
00020      C
00021      C  *****  DEFINITION OF VARIABLES *****
00022      C
00023      C  ALIP = MINOR AXIS OF THE ELLIPTICAL INLET LIP, INCHES
00024      C  BLIP = MAJOR AXIS OF THE ELLIPTICAL INLET LIP, INCHES
00025      C  FOR CIRCULAR ARC, ALIP = BLIP
00026      C  DIAM = DIAMETER OF INLET DUCT, INCHES
00027      C  FMACH= MACH NUMBER OF UNIFORM FLOW IN THE STRAIGHT DUCT SECTION
00028      C  XID = CUT-OFF RATIO OF MODE IN THE STRAIGHT DUCT SECTION
00029      C  XIBEL = CUT-OFF RATIO OF MODE IN THE ELLIPTIC BELLMOUTH AS THE
00030      C  MODE RELEASES AND RADIATES. USE THIS VALUE FOR RADIATION.
00031      C  XDARAD=VALUE OF X/A WHERE RADIATION RELEASES FROM BELLMOUTH
00032      C
00033      C  ** AN ITERATION IS REQUIRED TO DETERMINE XIRAD, START THE ITERATION
00034      C
00035      FMACH = FMACK
00036      IF(FMACK.LT.0.0) FMACH=-FMACK
00037      PI = 3.1415927
00038      N = 100
00039      FN = N
00040      RAD = DIAM/2.0
00041      BDA = BLIP/ALIP
00042      ADUCT = PI*DIAM**2/4.0
00043      FM1 = SQRT(1.0-FMACH**2)
00044      IF(XID.LE.1.0) GO TO 40
00045      C  ** IF THE MODE DOES NOT PROPAGATE IN THE DUCT, THE BELLMOUTH MAY
00046      C  STILL CUT THE MODE ON AND ALLOW PROPAGATION. SKIP DUCT
00047      C  CALCULATIONS BELOW IF DUCT CUT-OFF RATIO LESS THAN UNITY.
00048      QXI = 1.0-1.0/XID**2
00049      QNT = QXI/(1.0-QXI*FMACH**2)
00050      QNT = FM1*SQRT(QNT)
00051      C  ** ANGDUCT IS THE PROPAGATION ANGLE OF THE MODE IN THE STRAIGHT DUCT
00052      ANGDUCT = ACOS(QNT)
00053      TANANG = TAN(ANGDUCT)
00054      TANANG2 = TANANG**2
00055      XDATSQ = TANANG2/(TANANG2+BDA**2)

```

```

00056      XDAT = SQRT(XDATSQ)
00057      QNT = 1.0-XDATSQ
00058      RT = RAD+BLIP*(1.0-SQRT(QNT))
00059      XT = XDAT*ALIP
00060      DELX = (RT-FRAC*RAD)/TANANG
00061      XRAD = XT-DELX
00062      IF(DELX.GE.XT) THEN
00063      XIBEL = XID
00064      ANGDUCT = ANGDUCT*180.0/PI
00065      FMBEL = FMACH
00066      DBEL = DIAM
00067
00068      C
00069      C ** THE BELLMOUTH DOES NOT EFFECT THE RADIATION AT THIS CUTOFF RATIO
00070      GO TO 501
00071      END IF
00072      40 CONTINUE
00073      XDAPR = 0.0
00074      XDIFPR = XRAD
00075      ICALC = 0
00076      DO 50 I=1,N
00077      FI = I
00078      XDA = SQRT(FI/FN)
00079      IF(XDA.GE.1.0) XDA=0.9999
00080      SQXA = SQRT(1.0-XDA**2)
00081      QNT = BDA*XDA/SQXA
00082      C ***** ANGVAL IS THE SLOPE OF THE BELLMOUTH WALL AT THIS X/A (XDA)
00083      ANGVAL = ATAN(QNT)
00084      C
00085      C ** CALCULATE THE INCREASED FLOW AREA DUE TO THE BELLMOUTH AT THIS XDA
00086      C
00087      QRB = BLIP*(1.0-SQXA)
00088      RC = QRB/SIN(ANGVAL)
00089      AEX = 2.0*PI*RC*(RAD*ANGVAL+RC*(1.0-COS(ANGVAL)))
00090      C
00091      C ** AEX IS THE EXTRA FLOW AREA DUE TO THE BELLMOUTH AT THIS XDA
00092      ATOT = AEX+ADUCT
00093      FMBEL = FMACH*ADUCT/ATOT
00094      RADBEL = RAD+QRB
00095      FM1B = SQRT(1.0-FMBEL**2)
00096      XIBEL= XID*RADBEL*FM1/(RAD*FM1B)
00097      QXI = 1.0-1.0/XIBEL**2
00098      QNT = QXI/(1.0-QXI*FMBEL**2)
00099      C ***** CHECK IF MODE HAS STARTED PROPAGATING IN BELLMOUTH *****
00100      IF(QXI.LT.0.0) GO TO 50
00101      ICALC = ICALC+1
00102      QNT = FM1B*SQRT(QNT)
00103      C ***** ANGPROP IS THE ANGLE OF PROPAGATION OF THE MODE AT THIS XDA
00104      ANGPROP = ACOS(QNT)
00105      TANANG = TAN(ANGPROP)
00106      TANANG2 = TANANG**2
00107      XDATSQ = TANANG2/(TANANG2+BDA**2)
00108      XDAT = SQRT(XDATSQ)
00109      QNT = 1.0-XDATSQ
00110      RT = RAD+BLIP*(1.0-SQRT(QNT))
00111      XT = XDAT*ALIP
00112      DELX = (RT-FRAC*RADBEL)/TANANG
00113      XRAD = XT-DELX
00114      XDIF = XRAD-XDA*ALIP
00115      IF(XDIF.LE.0.0) GO TO 55

```

```

00116      XDIFPR = XDIF
00117      XDAPR = XDA
00118      50 CONTINUE
00119      IF(ICALC.EQ.0) THEN
00120      WRITE(*,100)
00121      100 FORMAT(' ***** ALERT ***** ALERT ***** ALERT *****')
00122      WRITE(*,101)
00123      101 FORMAT(' ** THIS MODE CAN NOT ESCAPE THE BELLMOUTH EXIT. PLEASE M
      LAKE MODIFICATIONS ** ')
00125      WRITE(*,100)
00126      XIBEL = .1
00127      GO TO 500
00128      END IF
00129      IF(ICALC.EQ.1) THEN
00130      C ***** MODE CUT-ON ACHIEVED AT BELLMOUTH EXIT *****
00131      XIBEL = 1.0001
00132      GO TO 500
00133      END IF
00134      55 CONTINUE
00135      IF(ICALC.EQ.1) THEN
00136      XDARAD = XDA
00137      GO TO 60
00138      END IF
00139      X2 = XDA
00140      X1 = XDAPR
00141      Y2 = XDIF
00142      Y1 = XDIFPR
00143      DY21 = ABS(Y2-Y1)
00144      IF(DY21.EQ.0.0) THEN
00145      XDARAD = XDAPR
00146      GO TO 60
00147      END IF
00148      XDARAD = (X1*Y2-X2*Y1)/(Y2-Y1)
00149      C
00150      C ** ITERATION DONE, CALCULATE OUTPUTS AT X/A = XDARAD
00151      C
00152      60 CONTINUE
00153      SQXA = SQRT(1.0-XDARAD**2)
00154      QNT = BDA*XDARAD/SQXA
00155      C ***** ANGVAL IS THE SLOPE OF THE BELLMOUTH WALL AT MODE RADIATION
      ANGVAL = ATAN(QNT)
00157      C
00158      C ** CALCULATE THE FINAL FLOW AREA INCREASE DUE TO THE BELLMOUTH
00159      C
00160      QRB = BLIP*(1.0-SQXA)
00161      RC = QRB/SIN(ANGVAL)
00162      AEX = 2.0*PI*RC*(RAD*ANGVAL+RC*(1.0-COS(ANGVAL)))
00163      C
00164      C ** AEX IS THE EXTRA FLOW AREA DUE TO THE BELLMOUTH AT THIS XDA
00165      ATOT = AEX+ADUCT
00166      FMBEL = FMACH*ADUCT/ATOT
00167      RADBEL = RAD+QRB
00168      FM1B = SQRT(1.0-FMBEL**2)
00169      XIBEL= XID*RADBEL*FM1/(RAD*FM1B)
00170      QXI = 1.0-1.0/XIBEL**2
00171      QNT = QXI/(1.0-QXI*FMBEL**2)
00172      QNT = FM1B*SQRT(QNT)
00173      IF(XID.GE.1.0) THEN
00174      QXI = 1.0-1.0/XID**2
00175      QNT = QXI/(1.0-QXI*FMACH**2)

```

```

00176      QNT = FM1*SQRT(QNT)
00177      C  ** "ANGDUCT" IS THE PROPAGATION ANGLE OF THE MODE FROM THE DUCT IF
      C      BELLMOUTH IS "NOT" CONSIDERED *****
      ANGDUCT = ACOS(QNT)*180.0/PI
00180      ELSE
00181      ANGDUCT = 90.0
00182      END IF
00183      500 CONTINUE
00184      DBEL = 2.0*RADBEL
00185      501 CONTINUE
00186      RETURN
00187      END

00001      C
00002      C *****
00003      C ***** END OF SUBROUTINE "LIPEF3" *****
00004      C *****
00005
00006      C
00007      C **** AFT FAN RADIATION SUBROUTINE, E. J. RICE, REVISED 02/19/1998
00008      C **** NORMALIZED PLANE WAVE RADIATION INCLUDED HERE.
00009      C ** THE FOLLOWING "BBRDCFEX" IS THE MAIN SUBROUTINE. ALL ARGUMENTS
00010      C      ARE INPUTS EXCEPT "DJET," "FMACH1," "NANGLE," "ANGLE," "SPL,"
00011      C      "SPLTL," "WATTS," "FMACHN," AND "COFMIN" AND WATTS
00012      C      RETURNED TO THE MAIN PROGRAM FOR PRINTING, FILE STORAGE ETC.
00013      C
      C ** THIS VERSION REMOVES THE P**2 OSCILLATIONS BEYOND THE PEAK. FOUR
00015      C      REGIONS (ETA AND CUT-OFF RATIO) DETERMINE PROPER APPROXIMATION FOR
      C      HIGH ANGLE RADIATION. DEVELOPED 01/30/98.
00017
00018      SUBROUTINE BBRDCFEX(TTOT,PTOT,TSUR,PSUR,HTRAT,
00019      1ANOZRAT,DISTANCE,ISIDELN,DDUCT,DJET,FMACHD,FMACH1,FMACH2,
00020      2NCOF,WATTSCOF,DELANG,ETAD,
00021      3NANGLE,ANGLE,SPL,SPLTL,WATTS,WATTRAN,FMACHN,COFMIN)
00022
00023      C
00024      DIMENSION ANGLE(200),SPL(200),SPLTL(200),WATTSCOF(200),
00025      1COFRAT(200),COFRATD(200),COFRATN(200),PSQTOT(200),
00026      2PSQTLOS(200),PSQRADT(200)
00027
00028      C
00029      C ** SUBROUTINE REQUIRED "CONOZ"
00030      C
00031      C ***** DEFINITION OF SUBROUTINE ARGUMENTS *****
00032      C
00033      C TTOT      TOTAL TEMPERATURE IN AFT FAN DUCT, (DEGREES RANKINE)
00034      C PTOT      TOTAL PRESSURE IN AFT FAN DUCT, (PSIA)
00035      C TSUR      TOTAL TEMPERATURE IN SURROUNDING MEDIUM, (DEGREES RANKINE)
00036      C PSUR      TOTAL PRESURE IN SURROUNDING MEDIUM, (PSIA)
00037      C HTRAT     AFT FAN DUCT HUB-TIP RATIO
00038      C ANOZRAT   (NOZZLE THROAT AREA)/(FAN DUCT AREA)
00039      C DISTANCE   RADIUS OF MICROPHONE ARRAY OR TRAVERSE, (FT.)
      C ISIDELN = 0 FOR CONSTANT RADIUS, = 1 FOR CONSTANT SIDELINE CALCULATION
      C DDUCT      AFT FAN DUCT OUTSIDE DIAMETER, (INCHES)
00042      C DJET      FINAL JET DIAMETER, (INCHES)
00043      C FMACHD     AFT FAN DUCT MACH NUMBER, POSITIVE FOR EXHAUST
00044      C FMACH1     FINAL JET MACH NUMBER
00045      C FMACH2     MACH NUMBER OF SURROUNDING MEDIUM
00046      C FMACHN     NOZZLE EXIT (THROAT) MACH NUMBER

```

```

00047 C NCOF      NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
00048 C WATTSCOF VECTOR WITH ACOUSTIC POWER FOR EACH BIN, (WATTS)
00049 C FREQ      FREQUENCY OF SOUND, (HERTZ)
      C DELANG   ANGLE INCREMENT DESIRED ON FAR-FIELD RADIAL TRAVERSE, (DEGREES)
00051 C NANGLE    NUMBER OF ANGLES USED IN CALCULATION, DETERMINED BY DELANG AND
      C          THE MAXIMUM ANGLE OF 180 DEGREES
00053 C ANGLE     VECTOR OF ANGLES ON THE FAR-FIELD TRAVERSE, CONVERTED TO
00054 C          THAT MEASURED FROM THE ENGINE "INLET" AXIS, (DEGREES)
00055 C SPL       THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
00056 C          "IGNORING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
00057 C           $2 \times 10^{(-5)}$  NEWTONS/METER**2
00058 C SPLTL    THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
      C          "INCLUDING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
      C           $2 \times 10^{(-5)}$  NEWTONS/METER**2
00061 C WATTS     SUM OF ACOUSTIC POWER IN ALL THE CUT-OFF RATIO BINS, (WATTS)
00062 C WATTRAN   SUM OF TRANSMITTED ACOUSTIC POWER IN ALL BINS, (WATTS)
00063 C COFMIN    THE MINIMUM CUT-OFF RATIO BELOW WHICH COMPLETE REFLECTION
00064 C          OCCURS DUE TO REFRACTION THROUGH THE SLIP LAYER
00065 C
00066          PI = 3.1415927
00067          QAFP = 1.0+0.328766*ETAD**1.702882
00068          AFPOWFAC = 1.741*(QAFP+1.274989*ETAD**2)/(PI*ETAD*QAFP)
00069
      C CALCULATE COEF. (PLANE WAVE) INTEGRATION OF RAD. DIRECTIVITY, MACH=0
      IF(ETAD.LT.1.0) THEN
00072          ACOEFPW = 1.733303+5.30259*ETAD**2.28937
00073          GO TO 14
00074          END IF
00075          ACOEFPW = 7.035893*ETAD**1.773669
00076      14 CONTINUE
00077 C
00078 C CALC. COEF. FOR INTEGRATION WITH AFT SHEAR LAYERS, PLANE WAVE
00079 C
00080          AFPOWFPW = (1.0+0.127683*ETAD)/(3.0+0.137590*ETAD)
00081          PSQCOEFP = ACOEFPW*AFPOWFPW*(1.0+2.6557*ETAD)/(1.0515+3.8508*ETAD)
00082          PSQCOEFP = 1.3704*PSQCOEFP
00083 C *****
00084
00085          FCOF = NCOF
00086          FCOFINV = 1./FCOF
00087          FCOFIND2 = 0.5/FCOF
00088 C ***** SET UP CUT-OFF RATIOS IN THE DUCT *****
00089          COFSQPR = 1.0
00090          DO 20 I=1,NCOF
00091              COFRATD(I) = SQRT(1.0/(COFSQPR-FCOFIND2))
00092              COFSQPR = COFSQPR-FCOFINV
00093      20 CONTINUE
00094
00095          TDUCT = TTOT/(1.0+0.2*FMACHD**2)
00096          CDUCT = 49.0421*SQRT(TDUCT)
00097
00098          QSUR = 1.0+0.2*FMACH2**2
00099          TSTS = TSUR/QSUR
00100          PSTS = PSUR/QSUR**3.5
00101          CSUR = 49.0421*SQRT(TSTS)
00102          RHOSUR = 144.0*PSTS/(53.3*TSTS)
00103
00104 C
00105 C ***** DETERMINE NOZZLE FLOW PROPERTIES *****
00106

```

```

00107      CALL CONOZ(TTOT,PTOT,PSTS,HTRAT,ANOZRAT,DDUCT,FMACHD,FMACH1,CJET,D
00108      1JET,TJET,PNOZ,DNOZ,CNOZ,FMACHN)
00109
00110      CRAT = CJET/CSUR
00111
00112      ETA = ETAD*DJET*CDUCT/(DDUCT*CJET)
00113      ETAN= ETAD*DNOZ*CDUCT/(DDUCT*CNOZ)
00114
00115      RATCFNOZ = DNOZ*CDUCT*SQRT(1.0-FMACHD**2)/(DDUCT*CNOZ*
00116      1SQRT(1.0-FMACHN**2))
00117
00118      RATCFJET = DJET*CDUCT*SQRT(1.0-FMACHD**2)/(DDUCT*CJET*
00119      1SQRT(1.0-FMACH1**2))
00120
00121
00122      FMSQ1 = FMACH1**2
00123      FM11 = 1.0-FMSQ1
00124      BETA1 = SQRT(FM11)
00125      FMSQ2 = FMACH2**2
00126  C
00127  C ***** ESTABLISH CUT-OFF LIMITS FOR COMPLETE REFLECTION
00128      COFMIN = 1.0
00129      IF(FMACH1.EQ.0.0) GO TO 15
00130
00131      CKM2 = 1.0-CRAT*FM11/FMACH1
00132      IF(FMACH2.LT.CKM2) THEN
00133      COSPHIL = -CRAT/(1.0+CRAT*FMACH1-FMACH2)
00134      PHIL = ACOS(COSPHIL)
00135      SINPHIL = SIN(PHIL)
00136      DEN = SQRT(1.0+FMSQ1+2.0*FMACH1*COSPHIL)
00137      COSPSIL = (COSPHIL+FMACH1)/DEN
00138      SINPSIL = SINPHIL/DEN
00139      COFMIN = SQRT(FM11+FMSQ1*COSPSIL**2)/(BETA1*SINPSIL)
00140  C      COFMIN = (1.0+FMACH1*COSPHIL)/(BETA1*SINPHIL)
00141      END IF
00142  15 CONTINUE
00143  C *****
00144
00145      NANGLE = 180.0/DELANG-1
00146      DO 5 I=1,NANGLE
00147      FI = I
00148      ANGLE(I) = FI*DELANG
00149      5 CONTINUE
00150
00151  C ***** PART OF EMPIRICAL COEFFICIENT TAKING PLACE OF RADICAL
00152  C ***** SQRT(1-1/COF**2) IN P**2 COEFICIENT WHICH WOULD NOT BE
00153  C ***** ANY GOOD AT CUT-OFF. FOR NON-PLANE WAVES.
00154
00155      ACOEF = 0.7/ETA
00156
00157  C ** RECALL, THE CUT-OFF RATIOS IN THE DUCT ARE INPUT HERE AS "COFRATD"
00158
00159  C ***** CALCULATE CUT-OFF RATIOS IN THE NOZZLE AND THE JET *****
00160      DO 22 I=1,NCOF
00161      COFRATN(I) = RATCFNOZ*COFRATD(I)
00162      COFRAT(I) = RATCFJET*COFRATD(I)
00163      22 CONTINUE
00164  C
00165  C ***** INITIALIZE P**2 AT EACH FAR-FIELD ANGLE *****
00166  C
00167      DO 10 I=1,NANGLE

```

```

00168      PSQRADT(I) = 0.0
00169      PSQTLOS(I) = 0.0
00170      PSQTOT(I) = 0.0
00171  10 CONTINUE
00172  C
00173  C ***** START LOOP ON CUT-OFF RATIO *****
00174  C
00175      POWCON = 8.36424*RHOSUR*CSUR
00176      WATTS = 0.0
00177      WATTRAN = 0.0
00178      DO 70 J=1,NCOF
00179      WATTS = WATTS+WATTSCOF(J)
00180  C
00181  C ** DETERMINE IF COFRAT IS WITHIN COMPLETE REFLECTION RANGE DUE TO
00182  C ** REFRACTION THROUGH THE SLIP LAYER OR IF COMPLETE REFLECTION
00183  C ** OCCURS AT THE NOZZLE THROAT
00184  C
00185      IF(COFRAT(J).LT.COFMIN.OR.COFRATN(J).LE.1.0) GO TO 70
00186
00187
00188  C *****
00189  C !!!!!!! VERY IMPORTANT! FOLLOWING DEFINES PLANE WAVE REGIME !!!!!
00190  C *****
00191
00192  C ***** DETERMINE IF PLANE WAVE RADIATION ALGORITHM SHOULD BE USED
00193  C ***** IF PLANE WAVE IPW = 1 (YES) FOR COFRATD>=3 OR LAST BIN
00194      IPW = 0
00195      IF(COFRATD(J).GE.3.0.OR.J.EQ.NCOF) IPW=1
00196
00197
00198
00199
00200  C ** CALCULATE TRANSMISSION LOSS AT NOZZLE THROAT CUT-OFF RATIO AND
00201  C ** FREQUENCY PARAMETER.
00202
00203      FMSQN = FMACHN**2
00204
00205      IF(IPW.EQ.1) THEN
00206
00207  C ***** IN PLANE-WAVE REGIME *****
00208
00209      X = 0.5*(PI*ETAN)**2
00210      RADRES = 1.0+X*EXP(-0.325226*X)-EXP(-0.101669*ETAN**5.7848)
00211      A = 0.023567
00212      Y = 0.5*PI**2*ETAN
00213      RADREC=EXP(-3.574331*ETAN**1.957292)*8.*
00214      1 ETAN/3.+A*Y**2/(1.+A*Y**3)
00215      QDEN = (1.0+FMACHN)**2*((RADRES+1.0)**2+RADREC**2)
00216      TLCF = 4.*(RADRES*(1.+FMSQN)+FMACHN*(RADRES**2+
00217      1 RADREC**2+1.))/QDEN
00218      GO TO 55
00219      END IF
00220
00221  C ***** IN NON-PLANE WAVE, RADIAL OR SPINNING MODE REGION *****
00222
00223      QF = PI*ETAN*(1.0-1.0/COFRATN(J))
00224      QF15SQ = (QF-1.5)**2
00225      RADREC = 1.135*EXP(-0.29*(QF+0.18)**2)
00226
00227      IF(QF.LE.1.5) THEN

```

```

00228      RADRES = 1.5*EXP(-0.2124*QF15SQ)
00229      GO TO 53
00230      END IF
00231
00232      RADRES = 1.0+0.5*EXP(-0.5338*QF15SQ)
00233 53 CONTINUE
00234
00235      TAU = SQRT(1.0-1.0/COFRATN(J)**2)
00236      TPM = TAU+FMACHN
00237      TTM = TAU*FMACHN
00238      QDEN = (RADRES*TPM+TTM+1.0)**2+(RADREC*TPM)**2
00239      QNUM = (RADRES+FMACHN)*(RADRES*FMACHN+1.0)+FMACHN*RADREC**2
00240      TLCF = 4.0*TAU*QNUM/QDEN
00241
00242 C ***** CALCULATE TRANSMITTED POWER *****
00243 C
00244 55 CONTINUE
00245      WATTRAN = WATTRAN+TLCF*WATTSCOF(J)
00246 C
00247 C ***** CALCULATE P**2 COEF. TO ACCOUNT FOR WATTS *****
00248 C      TO ACCOUNT FOR MIC. RADIUS, DIVIDE BY DISTANCE**2 AT PSQ BELOW.
00249 C
00250      POWCOEF = POWCON*WATTSCOF(J)
00251 C
00252 C ***** PROGRAM SPITS HERE. IT WILL CONTINUE ON FOR NON-PLANE
00253 C ***** WAVES AND WILL JUMP FOR THE PLANE WAVE
00254
00255      IF(IPW.EQ.1) GO TO 45
00256
00257 C ***** NOW IN "NON-PLANE" WAVE, RADIAL OR SPINNING MODE REGIME !!
00258
00259      COFBETIN = 1.0/(COFRAT(J)*BETA1)
00260      COFINV = 1.0/COFRAT(J)
00261      COFINVSQ = COFINV**2
00262      COFM1 = 1.0-COFINVSQ
00263      EP = SQRT(COFM1)
00264      GDEN = (1.0+EP)**2
00265
00266 C **** HERE IS REMAINDER OF EMPIRICAL COEFFICIENT USING "ACOE" ABOVE
00267
00268      A90 = 2.0*(ACOE+EP)/(ACOE+1.0)
00269
00270 C **** THEORETICAL NORMALIZATION COEFFICIENT WITH FLOW ATTACHED TO A90
00271 C ** INCLUDES INTEGRATED POWER NORMALIZATION "AFPOWFAC" (EMPIRICAL) **
00272
00273      PSQCOEF = AFPOWFAC*A90*(1.0-FMSQ1*COFM1)**1.5/BETA1
00274
00275      COSPK1 = BETA1*EP/SQRT(1.0-FMSQ1*COFM1)
00276      ANGPK1 = ACOS(COSPK1)
00277 C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 1, PSIPK1 (DEGREES)
00278      PSIPK1 = ANGPK1*180.0/PI
00279      SINPK1 = SIN(ANGPK1)
00280 C ***** PHASE VELOCITY VECTOR ANGLE FOR PEAK IN REGION 1 (JET)
00281      SIN2 = SINPK1**2
00282      COSPHI1 = -FMACH1*SIN2+COSPK1*SQRT(1.0-FMSQ1*SIN2)
00283 C
00284 C ***** PHI1 TO PHI2 ACROSS SLIP LAYER
00285 C ***** PHASE VELOCITY ANGLE IN REGION 2 (SURROUNDINGS)
00286
00287      COSPHI2 = COSPHI1/(CRAT+(CRAT*FMACH1-FMACH2)*COSPHI1)
00288

```

```

00289 C ***** PHASE TO GROUP VELOCITY ANGLES IN REGION 2 (SURROUNDINGS)
00290
00291     COSPSI2 = (COSPHI2+FMACH2)/SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00292     PSI2RAD = ACOS(COSPSI2)
00293     SINPSI2 = SIN(PSI2RAD)
00294
00295 C ***** ANGLE CHANGE ACOUSTIC POWER CORRECTION *****
00296
00297     FREFRCT = SINPK1/SINPSI2
00298
00299 C *****
00300
00301     Q22NUM = SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00302
00303     COSPSPK2 = (COSPHI2+FMACH2)/Q22NUM
00304     ANGPK2 = ACOS(COSPSPK2)
00305
00306 C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 2, PSIPK2 (DEGREES)
00307     PSIPK2 = ANGPK2*180.0/PI
00308
00309 C GROUP VELOCITY VECTOR ANGLE CHANGE, REG. 1 TO REG. 2, DELPSI (DEG.)
00310     DELPSI = PSIPK2-PSIPK1
00311 C *****
00312
00313     SIN2PK2 = SIN(PSIPK2*PI/180.0)
00314
00315 C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
00316     PSQPK = PSQCOEF*FREFRCT*ETA*COFRAT(J)/(2.0*BETA1)
00317 C *****
00318
00319
00320 C ***** START SORTING INTO REGIMES TO HANDLE ANGLES BEYOND PEAK ****
00321 C
00322     COF = COFRAT(J)
00323     IREG = 0
00324     ETAC1 = 0.6*BETA1/(1.0-COFINV)
00325     IF(ETA.GT.ETAC1) THEN
00326         IREG = 1
00327
00328 C ***** REGION 1, SLOPE AT ARG=PI/2 USED TO PROJECT BEYOND PEAK
00329
00330     EPS = 1.0/(BETA1*COF)+0.5/ETA
00331     EPSQ = EPS**2
00332     QNUM = 1.0+FMSQ1*EPSQ
00333     DEPDPSI = QNUM*SQRT(1.0-FM11*EPSQ)
00334     PSQRATC1 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00335     FDEN = CFBTINSQ-EPSQ
00336     DPSQDPSI = (FDEN+4.0*EPSQ)*PSQRATC1*DEPDPSI/(EPS*FDEN)
00337     SINPSIC1 = EPS/SQRT(QNUM)
00338     PSIC = ASIN(SINPSIC1)*180.0/PI
00339     AC = ALOG(PSQRATC1)
00340     BC = 0.8889*DPSQDPSI/PSQRATC1
00341     BC = BC*PI/180.0
00342     CC = -0.1781*BC
00343     GO TO 50
00344     END IF
00345
00346     ETAC2 = 0.6*BETA1*COF
00347     IF(ETA.GT.ETAC2) THEN
00348         IREG = 2

```

```

00349
00350 C ***** REGION 2, PSQratio AT ARG=-PI/2 USED TO FIT EXPONENTIAL FOR
00351 C           USE BEYOND PEAK
00352
00353     EPS = 1.0/(BETA1*COF)-0.5/ETA
00354     PSQRATC2 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
00355     EPSQ = EPS**2
00356     QDEN = 1.0+FMSQ1*EPSQ
00357     SINPSIC2 = EPS/SQRT(QDEN)
00358     PSIC2 = ASIN(SINPSIC2)*180.0/PI
00359     AC = ALOG(PSQRATC2)/(PSIPK1-PSIC2)**2
00360     GO TO 50
00361     END IF
00362
00363
00364
00365 C ***** REGION 3, LOW ETA REGION, PSIPK1>60 DEG. FIT EXPONENTIAL AT
00366 C 0.5*PSIPK1 FOR USE BEYOND PEAK. CODE IREG=3. EXPONENTIAL EQUATION
00367 C USED IN PSQ SUBROUTINE FOR PSI > PSIPK1.
00368
00369     IF(PSIPK1.GT.60.0) THEN
00370
00371     IREG = 3
00372
00373     ANGF = 0.5*PSIPK1
00374     ANGRAD = ANGF*PI/180.0
00375     SINF = SIN(ANGRAD)
00376     EPS = SINF/SQRT(1.0-FMSQ1*SINF**2)
00377     ARG = PI*ETA*(COFBETIN-EPS)
00378     SINARG = SIN(ARG)
00379     PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00380     PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00381     AC = ALOG(PSQRATC3)/(PSIPK1-ANGF)**2
00382     GO TO 50
00383     END IF
00384
00385 C ***** REGION 4, LOW ETA REGION, PSIPK1<60 DEG. FIT EXPONENTIAL AT
00386 C 80 deg. FOR USE BEYOND PEAK. CODE IREG=4. EXPONENTIAL EQUATION
00387 C USED IN PSQ SUBROUTINE FOR PSI>80 deg.
00388
00389     IREG = 4
00390
00391     ANGF = 80.0
00392     ANGRAD = ANGF*PI/180.0
00393     SINF = SIN(ANGRAD)
00394     EPS = SINF/SQRT(1.0-FMSQ1*SINF**2)
00395     ARG = PI*ETA*(COFBETIN-EPS)
00396     SINARG = SIN(ARG)
00397     PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
00398     PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
00399     AC = ALOG(PSQRATC3)/(PSIPK1-ANGF)**2
00400
00401     50 CONTINUE
00402 C
00403 C ***** FINISHED WITH 4 REGION DEFINITIONS, START PSQ CALCULATION
00404
00405     DO 25 I=1,NANGLE
00406     FI = I
00407     ANGDEG2 = ANGLE(I)
00408

```

```

00409      IF(ANGDEG2.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00410      PSQRADT(I) = 0.0
00411      PSQTOT(I) = 0.0
00412      PSQTLOS(I) = 0.0
00413      GO TO 25
00414      END IF
00415  C
00416      ANGDEG1 = ANGDEG2-DELPSI
00417      ANG = ANGDEG1
00418      ANGRAD1 = ANGDEG1*PI/180.0
00419      IF(ANGDEG1.LT.0.0) GO TO 25
00420      SINANG = SIN(ANGRAD1)
00421      COSANG = COS(ANGRAD1)
00422
00423      Q1DEN = SQRT(1.0-FMSQ1*SINANG**2)
00424      Q1 = SINANG/Q1DEN
00425      ARG = PI*ETA*(Q1-COFBETIN)
00426      SINSQNUM = (SIN(ARG))**2
00427      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00428      PSQRAT = 4.0*Q1/(BETA1*COFRAT(J)*(Q1+COFBETIN)**2)
00429
00430      PSQDEN = ARG**2
00431      ANGCK = PSIPK1+1.0
00432      IF(ANG.GT.ANGCK) GO TO 6
00433  C ***** CHECK FOR 0/0, USE LIMITING CASE IF INDETERMINATE *****
00434  C ***** DON'T CHECK IF ANGLE ABOVE PEAK ANGLE *****
00435      IF(PSQDEN.LT.1.E-06) GO TO 39
00436      6 CONTINUE
00437
00438      IF(ANG.LT.PSIPK1) GO TO 38
00439      IF(ANG.GE.PSIC.AND.IREG.EQ.1) THEN
00440
00441  C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00442  C ** USES SLOPE AT ARG = PI/2 TO EXTRAPOLATE TO HIGH ANGLES *****
00443      DANG = ANG-PSIC
00444      QEXP = AC+BC*DANG/(1.0+CC*DANG)
00445      IF(QEXP.LT.-20.) QEXP=-20.
00446
00447      PSQRAT = EXP(QEXP)
00448      GO TO 39
00449      END IF
00450
00451      IF(ANG.GE.PSIPK1.AND.IREG.EQ.2) THEN
00452
00453  C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
00454  C ** USES EXPONENTIAL FIT AT ARG = -PI/2 TO EXTRAPOLATE TO HIGH ANGLES
00455
00456      QEXP = AC*(ANG-PSIPK1)**2
00457      IF(QEXP.LT.-20.) QEXP=-20.
00458
00459      PSQRAT = EXP(QEXP)
00460      GO TO 39
00461      END IF
00462
00463      IF(ANG.GE.PSIPK1.AND.IREG.EQ.3) THEN
00464
00465  C ** ONE OF THE LOW FREQUENCY REGIONS, REGION 3. FOLLOWING ALLOWS
00466  C ** REASONABLE EXTRAPOLATION BEYOND 90 deg, STARTING AT PEAK.
00467  C ** USES EXPONENTIAL FIT AT PSI = PSIPK1 AND 0.5*PSIPK1. PSIPK1>60 deg
00468
00469      QEXP = AC*(ANG-PSIPK1)**2

```

```

00470      IF(QEXP.LT.-20.) QEXP=-20.
00471
00472      PSQRAT = EXP(QEXP)
00473      GO TO 39
00474      END IF
00475
00476 C  ** ONLY REGION LEFT, REGION 4, WITH PSIPK1<60 deg.
00477 C  ** ONE OF THE LOW FREQUENCY REGIONS. FOLLOWING ALLOWS REASONABLE
00478 C  ** EXTRAPOLATION BEYOND 90 deg, STARTING THIS TIME AT 80 deg.
00479 C  ** USES EXPONENTIAL FIT AT PSI = PSIPK1 AND 80 deg. PSIPK1<60 deg.
00480
00481      IF(ANG.LT.80.0) GO TO 38
00482
00483      QEXP = AC*(ANG-PSIPK1)**2
00484      IF(QEXP.LT.-20.) QEXP=-20.
00485
00486      PSQRAT = EXP(QEXP)
00487      GO TO 39
00488
00489      38 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00490      39 CONTINUE
00491
00492      PSQ = PSQRAT*PSQPK*GG
00493      PSQRAD = PSQ
00494 C
00495      RAD = DISTANCE
00496      IF(ISIDELN.EQ.1) THEN
00497      RAD = DISTANCE/SIN(ANGDEG2*PI/180.0)
00498      END IF
00499
00500      PSQ = PSQ/RAD**2
00501
00502 C
00503      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00504      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00505      PSQTL0S(I) = PSQTL0S(I)+POWCOEF*PSQ*TL0CF
00506
00507 C  ***** NOTE THAT A TRANSMISSION LOSS (TL0CF) HAS BEEN USED *****
00508
00509      25 CONTINUE
00510      26 CONTINUE
00511      GO TO 70
00512
00513
00514      45 CONTINUE
00515 C  ***** IN PLANE-WAVE CALCULATION PROCEDURE !!!!!!!!!!!!!!!!!!!!!!!
00516
00517      GDEN = 4.0
00518
00519      IF(FMACH1.EQ.FMACH2) THEN
00520      DELPSI = 0.0
00521      GO TO 27
00522      END IF
00523
00524      COSPK1 = 1.0
00525      ANGPK1 = 0.0
00526 C  ***** GROUP VELOCITY VECTOR ANGLE IN REGION 1, PSIPK1 (DEGREES)
00527      PSIPK1 = 0.0
00528      SINPK1 = 0.0
00529 C  ***** PHASE VELOCITY VECTOR ANGLE FOR PEAK IN REGION 1 (JET)
00530      SIN2 = SINPK1**2

```

```

00531      COSPHI1 = 1.0
00532      PHI1RAD = 0.0
00533      PHI1DEG = 0.0
00534      SINPHI1 = 0.0
00535  C
00536  C ***** PHASE VELOCITY ANGLE IN REGION 2 (SURROUNDINGS)
00537
00538      COSPHI2 = 1.0/(CRAT+CRAT*FMACH1-FMACH2)
00539      PHI2RAD = ACOS(COSPHI2)
00540      PHI2DEG = PHI2RAD*180.0/PI
00541      SINPHI2 = SIN(PHI2RAD)
00542      COSPSI2 = (COSPHI2+FMACH2)/SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00543      PSI2RAD = ACOS(COSPSI2)
00544      PSI2DEG = PSI2RAD*180.0/PI
00545      SINPSI2 = SIN(PSI2RAD)
00546
00547      Q22DEN = SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
00548
00549      COSPSPK2 = (COSPHI2+FMACH2)/Q22DEN
00550      ANGPK2 = ACOS(COSPSPK2)
00551      PSIPK2 = ANGPK2*180.0/PI
00552
00553  C GROUP VELOCITY VECTOR ANGLE CHANGE, REG. 1 TO REG. 2, DELPSI (DEG.)
00554      DELPSI = PSIPK2-PSIPK1
00555  C *****
00556
00557  C
00558      27 CONTINUE
00559
00560  C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
00561      PSQPK = 2.0*PSQCOEFP
00562  C *****
00563
00564  C ** PSI10 BELOW IS SMALLEST ANGLE WHERE PLANE WAVE RADIATION, P**2=0
00565  C ** IF PSI10 > 90, PSI10 = 90 IS USED
00566      SINPSI10 = 1.0/SQRT(ETA**2+FMACH1**2)
00567      IF(SINPSI10.LT.1.0) THEN
00568          ANG10 = ASIN(SINPSI10)
00569          PSI10 = ANG10*180.0/PI
00570          COSPSI10 = COS(ANG10)
00571          GO TO 28
00572      END IF
00573      ANG10 = PI/2.0
00574      PSI10 = 90.0
00575      SINPSI10 = 1.0
00576      COSPSI10 = 0.0
00577      28 CONTINUE
00578
00579  C ***** LOW ETA REGION. FIT EXPONENTIAL AT PSICRPL FOR BEYOND PEAK.
00580  C ***** IF PSICRPL < 90 CAN'T BE OBTAINED, USE PSICRPL = 90.
00581      ANGF = 90.0
00582      ETACRPL = 0.5*BETA1
00583      SINCRPL = 1.0/SQRT(4.0*ETA**2+FMACH1**2)
00584      IF(ETA.GT.ETACRPL) ANGF=ASIN(SINCRPL)*180./PI
00585      PSICRPL = ANGF
00586      ANGFRAD = ANGF*PI/180.0
00587      SINF = SIN(ANGFRAD)
00588      ARG = PI*ETA*SINF/SQRT(1.0-FMSQ1*SINF**2)
00589      SINARG = SIN(ARG)
00590      PSQRATPL = (SINARG/ARG)**2

```

```

00591      ACPL = ALOG(PSQRATPL)/ANGF**2
00592 C ***** ACPL IS THE AMPLITUDE OF THE EXPONENTIAL AT PSICRPL *****
00593 C ***** END LOW ETA REGION EXPONENTIAL FIT AT PSICRPL. *****
00594
00595 C CALCULATE P**2 AMPLITUDE CHANGE DUE TO ANGLE SHIFT FROM PSI = 0 TO
00596 C PSI = PSIPK2 FOR THE PLANE WAVE PASSING THROUGH THE JET SHEAR LAYER
00597
00598      PSQPKMUL = 1.0
00599      AREA1 = 1.0-COSPSI10
00600      AREA2 = 1.0+SINPSI2*SINPSI10-COSPSI2*COSPSI10
00601
00602      IF(PSIPK2.GT.0.0.AND.PSI10.LT.PSIPK2) THEN
00603      AREA2 = 2.0*SINPSI2*SINPSI10
00604      END IF
00605
00606      PSQPKMUL = AREA1/AREA2
00607      PSQPK = PSQPK*PSQPKMUL
00608
00609      CKPSI0 = -PSI10
00610      SUMPSQ = 0.0
00611
00612
00613      DO 40 I=1,NANGLE
00614      FI = I
00615      ANGDEG2 = ANGLE(I)
00616      IF(ANGDEG2.EQ.0.0.AND.ISIDELN.EQ.1) THEN
00617      PSQRADT(I) = 0.0
00618      PSQTOT(I) = 0.0
00619      PSQTLOS(I) = 0.0
00620      GO TO 40
00621      END IF
00622 C
00623      ANGRAD2 = ANGDEG2*PI/180.0
00624      ANGDEG1 = ANGDEG2-DELPSI
00625      ANGRAD1 = ANGDEG1*PI/180.0
00626
00627      IF(ANGDEG1.LT.CKPSI0) GO TO 40
00628
00629      SINANG = SIN(ANGRAD1)
00630      COSANG = COS(ANGRAD1)
00631
00632      Q1DEN = SQRT(1.0-FMSQ1*SINANG**2)
00633      Q1 = SINANG/Q1DEN
00634      ARG = PI*ETA*Q1
00635      SINSQNUM = (SIN(ARG))**2
00636      GG = (1.0+COSANG/Q1DEN)**2/GDEN
00637      PSQRAT = 1.0
00638
00639      PSQDEN = ARG**2
00640      IF(PSQDEN.LT.1.E-06.AND.ANGDEG1.LE.90.0) GO TO 49
00641
00642
00643 C ** LOW FREQUENCY REGION. FOLLOWING ALLOWS REASONABLE
00644 C ** EXTRAPOLATION BEYOND 90 deg, STARTING AT ARG = PI/2 IF ETA>ETAcrit
00645 C ** OR AT 90 deg. OTHERWISE. USES EXPONENTIAL FIT AT PSI = PSIPK1
00646 C ** AND PSI = PSIFIT.
00647
00648      IF(ANGDEG1.LT.PSICRPL) GO TO 48
00649
00650      QEXP = ACPL*(ANGDEG1)**2

```

```

00651      IF(QEXP.LT.-20.) QEXP=-20.
00652
00653      PSQRAT = EXP(QEXP)
00654      GO TO 49
00655
00656
00657      48 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
00658      49 CONTINUE
00659
00660      PSQ = PSQRAT*PSQPK*GG
00661
00662      PSQRAD = PSQ
00663  C
00664      RAD = DISTANCE
00665      IF(ISIDELN.EQ.1) THEN
00666      RAD = DISTANCE/SIN(ANGDEG2*PI/180.0)
00667      END IF
00668
00669      PSQ = PSQ/RAD**2
00670
00671  C
00672      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
00673      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
00674      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF
00675
00676  C ***** NOTE THAT A TRANSMISSION LOSS (TLCF) HAS BEEN USED *****
00677
00678      40 CONTINUE
00679      41 CONTINUE
00680  C
00681      70 CONTINUE
00682
00683      FNANGLE = NANGLE
00684      SUMWATT = 0.0
00685      DO 75 I=1,NANGLE
00686      ANGRAD = ANGLE(I)*PI/180.0
00687      SUMWATT = SUMWATT+PSQRADT(I)*SIN(ANGRAD)
00688
00689  C ** CONVERTING TO ANGLE FROM INLET AXIS FOR THIS AFT RADIATED NOISE **
00690
00691      ANGLE(I) = 180.0-ANGLE(I)
00692      IF(PSQTOT(I).LT.4.E-08) THEN
00693      SPLTL(I) = 20.0
00694      SPL(I) = 20.0
00695      GO TO 75
00696      END IF
00697      SPLTL(I) = 10.0*ALOG10(PSQTLOS(I))+93.9794
00698      SPL(I) = 10.0*ALOG10(PSQTOT(I))+93.9794
00699      75 CONTINUE
00700
00701      WATTINT = SUMWATT*PI/(POWCON*FNANGLE)
00702      SPLDIF = 10.0*ALOG10(WATTS/WATTINT)
00703
00704      DO 80 I=1,NANGLE
00705      SPLTL(I) = SPLTL(I)+SPLDIF
00706      SPL(I) = SPL(I)+SPLDIF
00707      80 CONTINUE
00708
00709
00710      RETURN
00711      END

```

```

00001  C
00002  C *****
00003  C ***** END OF MAIN SUBROUTINE "BBRDCFEX" *****
00004  C ***** ALTERED 02/19/1998, E. J. RICE *****
00005  C *****
00006  C
00007  C
00008  C*****
00009  C ** SUBROUTINE FOR CALC NOZZLE CONDITIONS, FINAL JET MACH NUMBER,
      C VELOCITY, AND DIAMETER. ** NOTE ** OUTER DIAM CHANGES FOR NOZZLE.
      C
00012      SUBROUTINE CONOZ(TTOT,PTOT,PSTS,HTRAT,ANOZRAT,DDUCT,FMACHD,FMACH1
00013      1,CJET,DJET,TJET,PNOZ,DNOZ,CNOZ,FMACHN)
00014
00015  C
00016      PI = 3.1415927
00017      QM = 1.0+0.2*FMACHD**2
00018      DIN = DDUCT*HTRAT
00019      ADUCT = PI*(DDUCT**2-DIN**2)/4.0
00020      ANOZ = ADUCT*ANOZRAT
00021      DNOZ = SQRT(4.0*ANOZ/PI+DIN**2)
00022      TDUCT = TTOT/QM
00023      VSOND = 49.0421*SQRT(TDUCT)
00024      VDUCT = FMACHD*VSOND
00025      RHOT = 144.0*PTOT/(53.3*TTOT)
00026      RHOD = RHOT/QM**2.5
00027      FMASS = RHOD*ADUCT*VDUCT
00028  C ***** NOTE - AREAS ARE IN SQUARE INCHES *****
00029  C
00030  C ***** SOLVE FOR NOZZLE MACH NUMBER *****
00031  C
00032      QQDUCT = FMACHD/QM**3/ANOZRAT
00033      FN = 1.0
00034      DIFP = FN/(1.0+0.2*FN**2)**3-QQDUCT
00035      FNP = 1.0
00036
00037      DO 10 I=1,50
00038      FN = 0.975*FN
00039      DIF = FN/(1.0+0.2*FN**2)**3-QQDUCT
00040  C      WRITE(3,100) I,FN,DIF
00041  C 100 FORMAT(/,' I=',I2,' FN=',F7.4,' DIF=',1PE9.2)
00042      IF(DIF.LE.0.0) GO TO 12
00043      FNP = FN
00044      DIFP = DIF
00045      10 CONTINUE
00046      FN = QQDUCT
00047      111 DO 11 I=1,10
00048      FN = QQDUCT*(1.0+0.2*FN**2)**3
00049  C      WRITE(3,102) I,FN
00050  C 102 FORMAT(/,' REFINED, I=',I2,' FN=',F7.4)
00051
00052      11 CONTINUE
00053      GO TO 14
00054      12 CONTINUE
00055      FN = (DIFP*FN-DIF*FNP)/(DIFP-DIF)
00056  C      WRITE(3,101) FN
00057  C 101 FORMAT(/,' INTERPOLATED FN =',F7.4)
00058

```

```

00059      GO TO 111
00060      14 CONTINUE
00061      FMACHN = FN
00062      C      WRITE(3,103) FMACHN
00063      C      103 FORMAT(/,' FINAL ***** FMACHN = ',F7.4)
00064
00065      C      ***** "FMACH1" FINAL JET VELOCITY *****
00066
00067      FMACH1 = SQRT(5.0*((PTOT/PSTS)**(2./7.)-1.0))
00068      QNTJ = 1.0+0.2*FMACH1**2
00069      TJET = TTOT/QNTJ
00070      RHOJ = RHOT/QNTJ**2.5
00071      CJET = 49.0421*SQRT(TJET)
00072      VJET = CJET*FMACH1
00073      AJET = FMACH1/(RHOJ*VJET)
00074      DJET = SQRT(4.0*AJET/PI+DIN**2)
00075
00076      C ** FOLLOWING NOT USED HERE BUT MIGHT BE HANDY FOR LATER USE *****
00077      C ** CHECK NOZZLE THROAT PRESSURE VRS. SURROUNDING AMBIENT PRESSURE *
00078      QMN = 1.0/(1.0+0.2*FMACHN**2)
00079      PNOZ = PTOT*QMN**3.5
00080      TNOZ = TTOT*QMN
00081      CNOZ = 49.0421*SQRT(TNOZ)
00082      RETURN
00083      END

00001      C
00002      C *****
00003      C ***** END OF SUBROUTINE "CONOZ" *****
00004      C *****
00005
00006      FUNCTION SINH(X)
00007      SINH = (EXP(X)-EXP(-X))/2.
00008      RETURN
00009      END

00001      FUNCTION TANH(X)
00002      TX = 2.*X
00003      ETX = EXP(-TX)
00004      TANH = (1.-ETX)/(1.+ETX)
00005      RETURN
00006      END

00001      FUNCTION ARCSECH(X)
00002      RX = 1./X
00003      TRM = SQRT(RX**2-1.)
00004      ARCSECH = ALOG(RX+TRM)
00005      RETURN
00006      END

00001      FUNCTION COTH(X)
00002      COTH = 1./TANH(X)
00003      RETURN
00004      END

```

```

00001      FUNCTION U1(X)
00002      X3      = X**3
00003      U1      = (3.*X-5.*X3)/24.
00004      RETURN
00005      END

00001      FUNCTION U2(X)
00002      X2      = X**2
00003      X4      = X2**2
00004      X6      = X2*X4
00005      U2      = (81.*X2-462.*X4+385.*X6)/1152.
00006      RETURN
00007      END

00001      FUNCTION V1(X)
00002      X3      = X**3
00003      V1      = (-9.*X+7.*X3)/24.
00004      RETURN
00005      END

00001      FUNCTION V2(X)
00002      X2      = X**2
00003      X4      = X2**2
00004      X6      = X2*X4
00005      V2      = (-135.*X2+594.*X4-455.*X6)/1152.
00006      RETURN
00007      END

00001      SUBROUTINE ABESJ ( ARG,ORD,BESJ )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESJ NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI      = 3.14159265
00005      ORD2    = ORD**2
00006      SECHAL  = ARG/ORD
00007      AL      = ARCSECH( SECHAL )
00008      TANHALL = TANH( AL )
00009      COTHALL  = 1./TANHALL
00010      RNUMJ    = EXP( ORD*( TANHALL-AL ) )
00011      DENJ     = SQRT( 2.*PI*ORD*TANHALL )
00012      BESJ     = (RNUMJ/DENJ)*(1.+U1( COTHALL )/ORD+U2( COTHALL )/ORD2)
00013 100    CONTINUE
00014      RETURN
00015      END

00001      SUBROUTINE ABESY ( ARG,ORD,BESY,RBESY )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESY NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI      = 3.14159265
00005      ORD2    = ORD**2
00006      SECHAL  = ARG/ORD
00007      AL      = ARCSECH( SECHAL )
00008      TANHALL = TANH( AL )
00009      COTHALL  = 1./TANHALL
00010      RNUMJ    = EXP( ORD*( TANHALL-AL ) )
00011      DENJ     = SQRT( 2.*PI*ORD*TANHALL )

```

```

00012      BESJ      = (RNUMJ/DENJ)*(1.+U1(COTHAL)/ORD+U2(COTHAL)/ORD2)
00013      ABESJ      = ABS(BESJ)
00014      IF ( ABESJ.LE.1.E-36 ) GO TO 90
00015      RNUMY      = -1./RNUMJ
00016      DENY        = 0.5*DENJ
00017      BESY        = (RNUMY/DENY)*(1.-U1(COTHAL)/ORD+U2(COTHAL)/ORD2)
00018      RBESY       = 1./BESY
00019      GO TO 100
00020      90    BESY    = -1.E+36
00021      RBESY      = 0.00
00022      100   CONTINUE
00023      RETURN
00024      END

```

```

00001      SUBROUTINE ABESJD ( ARG,ORD,BESJD )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESJD NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI          = 3.14159265
00005      ORD2         = ORD**2
00006      SECHAL      = ARG/ORD
00007      AL          = ARCSECH(SECHAL)
00008      TANHAL      = TANH(AL)
00009      COTHAL      = 1./TANHAL
00010      RNUMJ       = SQRT(SINH(2.*AL))*EXP(ORD*(TANHAL-AL))
00011      DENJ        = SQRT(4.*PI*ORD)
00012      BESJD       = (RNUMJ/DENJ)*(1.+V1(COTHAL)/ORD+V2(COTHAL)/ORD2)
00013      100   CONTINUE
00014      RETURN
00015      END

```

```

00001      SUBROUTINE ARBESYD ( ARG,ORD,RBESYD )
00002      IF ( ARG.GE.ORD ) PRINT *, 'ARG.GE.ORD-ABESJY NOT APPLICABLE'
00003      IF ( ARG.GE.ORD ) GO TO 100
00004      PI          = 3.14159265
00005      ORD2         = ORD**2
00006      SECHAL      = ARG/ORD
00007      AL          = ARCSECH(SECHAL)
00008      TANHAL      = TANH(AL)
00009      COTHAL      = 1./TANHAL
00010      RNUMJ       = SQRT(SINH(2.*AL))*EXP(ORD*(TANHAL-AL))
00011      DENJ        = SQRT(4.*PI*ORD)
00012      BESJD       = (RNUMJ/DENJ)*(1.+V1(COTHAL)/ORD+V2(COTHAL)/ORD2)
00013      ABESJD      = ABS(BESJD)
00014      IF ( ABESJD.LE.1.E-36 ) GO TO 90
00015      RNUMY       = SQRT(SINH(2.*AL))*EXP(ORD*(AL-TANHAL))
00016      DENY        = 0.5*DENJ
00017      BESYD       = (RNUMY/DENY)*(1.-V1(COTHAL)/ORD+V2(COTHAL)/ORD2)
00018      RBESYD      = 1./BESYD
00019      GO TO 100
00020      90    BESYD    = 1.E+36
00021      RBESYD     = 0.00
00022      100   CONTINUE
00023      RETURN
00024      END

```

```

00001      SUBROUTINE ABESYR ( A1,A2,ORD,RES )
00002      IF ( A1.GE.ORD ) PRINT *, '1ST ARG.GE.ORD-ABESYR ',

```

```

00003      &                                'NOT APPLICABLE'
00004      IF ( A1.GE.ORD ) GO TO 100
00005      IF ( A2.GE.ORD ) PRINT *, '2ND ARG.GE.ORD-ABESYR ',
00006      &                                'NOT APPLICABLE'
00007      IF ( A2.GE.ORD ) GO TO 100
00008      PI      = 3.14159265
00009      ORD2     = ORD**2
00010      SECHAL1  = A1/ORD
00011      AL1     = ARCSECH( SECHAL1 )
00012      TANHAL1  = TANH( AL1 )
00013      COTHAL1  = 1./TANHAL1
00014      SECHAL2  = A2/ORD
00015      AL2     = ARCSECH( SECHAL2 )
00016      TANHAL2  = TANH( AL2 )
00017      COTHAL2  = 1./TANHAL2
00018      SINHAL2  = SINH( 2.*AL2 )
00019      EXPON    = -ORD*( ( AL2-TANHAL2 ) - ( AL1-TANHAL1 ) )
00020      DEN      = -SQRT( TANHAL1*SINH2AL2/2. )
00021      SRAT     = ( 1.-U1( COTHAL1 )/ORD+U2( COTHAL1 )/ORD2 ) /
00022      &       ( 1.-V1( COTHAL2 )/ORD+V2( COTHAL2 )/ORD2 )
00023      RES      = EXP( EXPON ) *SRAT/DEN
00024      100     CONTINUE
00025      RETURN
00026      END

```

2.2.3.5 Sample Run Program STATOR

Current code name is SDIRFIN. To run the program first move to the directory that the code executable file resides in, then invoke the program by typing after the system prompt. From a Unix system type “program_name.” From a VAX system type “run program_name.” A typical run will look like this:

```

prompt> sdirfin
Enter input file name : bbn_input.dat
Enter output file name : stator_power.dat
Enter 2nd output file name: stator_debug.dat
Enter spl plot output file name: stator_spl.dat
prompt>

```

2.2.3.6 Sample Input File

```

$Input
RPM = 9782.2
RHO = 0.079005
DTIP =      22
HTR = 0.43
NBLADE =     22
NSTR =      12
NVANE =     54
GAM = 1.4
KASE =       1
LINLET =    0.99
LEXIT =    1.99
IABSOR =     0
NBSTD =     22
SCLOPTR =    4

```

```

SCLOPTS =      2
NHM = 10
BW =          0
NF =         11
NTOBNI =       7
NCOF =        15
RADMIC =       20
ISIDELN =      1
ALIP =        5.558
BLIP =        1.05
MACHS =        0.2
ANOZRAT =     0.6875
ETAFAN =      0.920
DELANG =     10.0
ITL = 1
TOBN =        36  37  38  39  40  41  42  43  44  45  46
SEMA =        0.418,0.422,0.428,0.436,0.444,0.450,0.453,0.454,0.451,0.447,0.443,0.438,
SPERC =        4.77, 5.23, 5.77, 6.44, 7.32, 8.54,10.32, 9.00, 9.41,.63, 6.77, 6.77,
SINCDR =       11.84,20.04,24.57,29.37,32.30,32.08,34.45,35.49,34.85,32.70,30.83,27.32,
SATIR =        0.095,0.002,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,
SATIS =        0.081,0.021,0.018,0.021,0.023,0.023,0.022,0.018,0.016,.013,0.012,0.024,
SELINR =       12*0.110,
SELINS =       0.128,0.063,0.064,0.067,0.072,0.075,0.092,0.091,0.126,0.132,0.119,0.106,
SEMT =         0.81608 0.78311 0.74889 0.71207 0.67321 0.63208,
               .58778 0.53878 0.50051 0.46351 0.43564 0.41233
SCO = 1138.1 1132.5 1130.2 1129.9 1129.2 1127.8
               .1 1124.4 1122.8 1121.3 1120.3 1119
STHETA =      0.28304 0.29918 0.32337 0.35848 0.39996 0.44971
               .51579 0.61069 0.70143 0.81039 0.90401 0.99027
SAXSP =       1.4713 1.4837 1.5001 1.5247 1.5563 1.5928
               .6373 1.6972 1.7526 1.8172 1.8753 1.9333
SROTCD =      0.06511 0.036664 0.016138 0.01422 0.012293 0.0066824
               .0019007 0.0026553 0.0013424 0.0013326 0.0069932 0.01341
SSADIN =      12.289 11.351 11.008 11.028 11.187 11.454
               .877 12.526 13.033 13.539 13.868 14.172
SCHDR =       4.227 4.124 4.020 3.911 3.794 3.675
               .546 3.399 3.285 3.176 3.096 3.028
SCHDS =       1.839 1.886 1.921 1.947 1.964 1.970
               .964 1.944 1.919 1.888 1.861 1.837
STPRIN =      1.344 1.363 1.365 1.355 1.339 1.321
               .299 1.271 1.252 1.235 1.223 1.214
SDIA = 21.988 20.939 19.844 18.699 17.494 16.214
               .829 13.298 12.101 10.962 10.126 9.440
SSTATCD =     0.02 0.02 0.02 0.02 0.02 0.02
               .02 0.02 0.02 0.02 0.02 0.02
SINCDS =      1 1 1 1 1 1 1 1 1 1 1 1
SATIW =       0.1 0.1 0.1 0.1 0.1 0.1
               .1 0.1 0.1 0.1 0.1 0.1
SCONTR =      1 1 1 1 1 1 1 1 1 1 1 1
SCONTS =      1 1 1 1 1 1 1 1 1 1 1 1
SCONTW =      1 1 1 1 1 1 1 1 1 1 1 1
SELINW =      0.1 0.1 0.1 0.1 0.1 0.1
               .1 0.1 0.1 0.1 0.1 0.1
SSCLR =       1 1 1 1 1 1 1 1 1 1 1 1
SSCLS =       1 1 1 1 1 1 1 1 1 1 1 1
SSCLW =       1 1 1 1 1 1 1 1 1 1 1 1
STVELR =      1 1 1 1 1 1 1 1 1 1 1 1
STVELS =      1 1 1 1 1 1 1 1 1 1 1 1
STVELW =      1 1 1 1 1 1 1 1 1 1 1 1
$

```

2.2.3.7 Sample Output Files

The STATOR power and directivity output files are shown in the following sections. The debug output file is not listed.

2.2.3.7.1 Power Output File from STATOR

```
HARD WALL ASSUMED
MICROPHONE IS ON A SIDELINE
MICROPHONE DISTANCE IN FEET IS =    20.00000
MACH NUMBER OF SURROUNDING MEDIUM =    0.2000000
ESTIMATED FAN ADIABATIC EFFICIENCY =    0.9200000

PROGRAM *** STATIN ***

RESPONSE OF AN ISOLATED OGV LIKE STATOR
TO INGESTION OF INLET TURBULENCE

CASE NUMBER    1    OF    1

***** STRIP AREA NUMBER    1    WWND = 1.0000

      EMA      EMTIP      TI      SINCD      CONTR      L/SSTD
      0.418      0.816      0.0810      1.00      1.000      0.13

      GAM      RHO      C      SDIA      SPERC      TPR
      1.400      0.0790      1138.      21.988      4.770      1.344

      RPM      NB      NBSTD      HTR      DTIP      CHDR
      9782.2      22      22      0.430      22.000      4.227

      EMR      RSCAL      RVEL      ELT      TIT      AR
      0.418      1.00      1.0000      0.128      0.0810      3.409

                        NB      NS
                        22      54

CDROTOR =    6.5109998E-02
INLET LENGTH/TIP DIAMETER =    0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT=    2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

      FREQUENCY      DIPOLE      QUADRUPOLE      TOTAL
      HERTZ      F/BPF      INLET EXHAUST      INLET EXHAUST      INLET EXHAUST
      3981      1.1099      93.4      108.2      97.4      109.2      98.9      111.7
      5011      1.3973      94.2      109.3      100.3      112.6      101.2      114.3
      6309      1.7591      91.7      107.7      100.4      113.4      100.9      114.4
      7943      2.2146      86.6      102.7      99.0      112.1      99.2      112.6
      10000      2.7880      84.4      100.2      100.0      112.9      100.1      113.1
      12589      3.5099      84.5      101.1      99.4      112.4      99.5      112.7
      15848      4.4187      82.4      99.5      98.5      111.8      98.6      112.0
      19952      5.5628      76.8      94.5      96.7      110.7      96.8      110.8
      25118      7.0031      71.9      90.6      93.2      108.0      93.2      108.1
      31622      8.8164      63.1      84.3      87.2      104.9      87.3      104.9
      39810      11.0992      61.5      80.5      86.0      101.8      86.0      101.9
```

***** STRIP AREA NUMBER 2 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.422	0.783	0.0210	1.00	1.000	0.06
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1133.	20.939	5.230	1.363
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	4.124
EMR	RSCAL	RVEL	ELT	TIT	AR
0.422	1.00	1.0000	0.063	0.0210	3.324
		NB	NS		
		22	54		

CDROTOR = 3.6664002E-02
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	79.6	93.7	85.3	96.6	86.3	98.4
5011	1.3973	79.1	93.5	87.4	98.6	88.0	99.8
6309	1.7591	78.1	93.2	90.3	101.8	90.6	102.4
7943	2.2146	76.2	91.6	92.1	103.7	92.2	104.0
10000	2.7880	72.5	88.6	90.9	103.1	91.0	103.2
12589	3.5099	74.0	89.4	91.9	103.8	92.0	104.0
15848	4.4187	71.5	88.0	90.9	103.2	91.0	103.3
19952	5.5628	66.1	82.7	89.6	102.1	89.6	102.2
25118	7.0031	61.9	79.7	86.4	100.1	86.4	100.2
31622	8.8164	53.6	73.0	81.0	96.4	81.0	96.4
39810	11.0992	51.8	69.3	79.4	93.7	79.4	93.7

***** STRIP AREA NUMBER 3 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.428	0.749	0.0180	1.00	1.000	0.06
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1130.	19.844	5.770	1.365
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	4.020
EMR	RSCAL	RVEL	ELT	TIT	AR
0.428	1.00	1.0000	0.064	0.0180	3.264
		NB	NS		
		22	54		

CDROTOR = 1.6138000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	80.3	95.0	87.3	98.5	88.1	100.1
5011	1.3973	78.1	93.4	87.3	98.7	87.8	99.8
6309	1.7591	77.1	91.1	88.8	99.3	89.1	99.9
7943	2.2146	74.4	88.8	90.8	101.3	90.9	101.6
10000	2.7880	74.2	89.0	93.1	103.6	93.1	103.7
12589	3.5099	74.3	89.4	92.7	103.6	92.8	103.8
15848	4.4187	71.2	86.9	91.0	102.5	91.1	102.6
19952	5.5628	66.7	82.5	90.8	102.1	90.8	102.2
25118	7.0031	62.5	79.5	87.3	99.8	87.4	99.8
31622	8.8164	55.4	73.2	83.1	96.6	83.1	96.6
39810	11.0992	52.1	69.3	80.2	94.1	80.2	94.1

***** STRIP AREA NUMBER 4 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.436	0.712	0.0210	1.00	1.000	0.07

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1130.	18.699	6.440	1.355

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.911

EMR	RSCAL	RVEL	ELT	TIT	AR
0.436	1.00	1.0000	0.067	0.0210	3.220

NB	NS
22	54

CDROTOR = 1.4220000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	84.1	97.8	92.2	102.2	92.9	103.6
5011	1.3973	83.8	97.2	94.0	103.9	94.4	104.8
6309	1.7591	81.4	95.6	94.5	104.8	94.7	105.3
7943	2.2146	78.0	92.1	94.7	104.9	94.8	105.2
10000	2.7880	75.5	90.0	94.1	104.3	94.2	104.5
12589	3.5099	76.3	90.8	94.5	104.6	94.6	104.8
15848	4.4187	75.1	89.7	95.8	105.8	95.9	105.9
19952	5.5628	70.8	85.9	95.4	105.7	95.4	105.7
25118	7.0031	66.7	83.0	91.9	103.3	91.9	103.3
31622	8.8164	60.3	77.0	88.3	100.5	88.3	100.5
39810	11.0992	55.9	72.7	84.9	97.4	84.9	97.4

***** STRIP AREA NUMBER 5 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.444	0.673	0.0230	1.00	1.000	0.07
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1129.	17.494	7.320	1.339
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.794
EMR	RSCAL	RVEL	ELT	TIT	AR
0.444	1.00	1.0000	0.072	0.0230	3.192
		NB	NS		
		22	54		

CDROTOR = 1.2293000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	87.1	99.5	95.1	103.9	95.7	105.3
5011	1.3973	86.7	99.8	97.1	106.2	97.5	107.1
6309	1.7591	85.1	98.1	98.9	108.0	99.1	108.5
7943	2.2146	82.5	96.0	100.2	109.4	100.3	109.6
10000	2.7880	81.3	95.0	101.0	110.3	101.0	110.4
12589	3.5099	81.9	96.0	101.2	110.6	101.2	110.8
15848	4.4187	79.3	93.8	100.6	110.3	100.7	110.4
19952	5.5628	74.3	89.2	99.2	109.2	99.3	109.2
25118	7.0031	71.2	86.7	96.7	107.1	96.7	107.2
31622	8.8164	64.9	81.1	93.1	104.3	93.1	104.4
39810	11.0992	60.2	76.5	89.9	101.5	89.9	101.5

***** STRIP AREA NUMBER 6 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.450	0.632	0.0230	1.00	1.000	0.08
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1128.	16.214	8.540	1.321
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.675
EMR	RSCAL	RVEL	ELT	TIT	AR
0.450	1.00	1.0000	0.075	0.0230	3.183
		NB	NS		
		22	54		

CDROTOR = 6.6824001E-03
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	87.2	99.5	94.8	103.0	95.5	104.6
5011	1.3973	86.5	99.1	97.0	105.3	97.4	106.2
6309	1.7591	85.2	98.1	99.0	107.3	99.1	107.8
7943	2.2146	82.9	95.9	100.6	109.0	100.6	109.2
10000	2.7880	82.1	95.5	101.7	110.2	101.7	110.3
12589	3.5099	83.3	96.8	102.5	111.1	102.5	111.2
15848	4.4187	81.5	95.3	102.9	111.6	103.0	111.7
19952	5.5628	78.2	92.3	103.1	111.9	103.2	112.0
25118	7.0031	77.4	91.7	103.1	112.0	103.2	112.1
31622	8.8164	74.6	89.1	103.0	112.0	103.0	112.0
39810	11.0992	72.3	87.0	102.7	111.8	102.7	111.8

***** STRIP AREA NUMBER 7 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.453	0.588	0.0220	1.00	1.000	0.09

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1126.	14.829	10.320	1.299

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.546

EMR	RSCAL	RVEL	ELT	TIT	AR
0.453	1.00	1.0000	0.092	0.0220	3.192

NB	NS
22	54

CDROTOR = 1.9007000E-03
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	86.3	98.4	93.5	101.0	94.2	102.9
5011	1.3973	86.5	98.4	95.7	103.1	96.2	104.4
6309	1.7591	84.6	96.8	97.2	104.7	97.4	105.3
7943	2.2146	82.0	94.5	98.4	105.9	98.5	106.2
10000	2.7880	80.9	94.0	99.1	106.7	99.2	106.9
12589	3.5099	81.9	95.1	99.5	107.2	99.6	107.4
15848	4.4187	79.6	93.3	99.6	107.3	99.6	107.5
19952	5.5628	75.9	90.2	99.4	107.2	99.4	107.3
25118	7.0031	74.9	89.9	99.0	107.0	99.0	107.1
31622	8.8164	71.7	87.1	98.5	106.6	98.6	106.6
39810	11.0992	69.2	85.0	98.0	106.0	98.0	106.1

***** STRIP AREA NUMBER 8 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.454	0.539	0.0180	1.00	1.000	0.09
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1124.	13.298	9.000	1.271
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.399
EMR	RSCAL	RVEL	ELT	TIT	AR
0.454	1.00	1.0000	0.091	0.0180	3.225
		NB	NS		
		22	54		

CDROTOR = 2.6553001E-03
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	80.9	92.0	87.3	94.0	88.2	96.1
5011	1.3973	80.3	92.0	89.3	96.0	89.8	97.4
6309	1.7591	78.6	90.4	91.1	97.8	91.4	98.6
7943	2.2146	76.4	88.1	92.4	99.2	92.5	99.5
10000	2.7880	75.7	87.9	93.4	100.3	93.5	100.5
12589	3.5099	76.9	88.9	94.0	100.9	94.1	101.1
15848	4.4187	74.3	86.8	94.1	101.1	94.2	101.3
19952	5.5628	71.1	83.8	94.1	101.2	94.1	101.3
25118	7.0031	69.9	82.7	93.8	101.0	93.8	101.1
31622	8.8164	66.8	80.0	93.4	100.6	93.4	100.7
39810	11.0992	64.3	77.8	92.8	100.1	92.8	100.1

***** STRIP AREA NUMBER 9 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.451	0.501	0.0160	1.00	1.000	0.13
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1123.	12.101	9.410	1.252
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.285
EMR	RSCAL	RVEL	ELT	TIT	AR
0.451	1.00	1.0000	0.126	0.0160	3.267
		NB	NS		
		22	54		

CDROTOR = 1.3424000E-03
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	79.2	90.2	85.3	91.7	86.3	94.0
5011	1.3973	78.4	89.3	86.6	92.9	87.2	94.5
6309	1.7591	76.8	88.2	87.9	94.2	88.2	95.1
7943	2.2146	74.3	85.9	89.0	95.2	89.1	95.7
10000	2.7880	74.0	85.3	89.8	96.0	89.9	96.3
12589	3.5099	74.6	86.1	89.9	96.3	90.0	96.7
15848	4.4187	71.8	83.8	89.9	96.4	89.9	96.6
19952	5.5628	68.6	80.6	89.8	96.4	89.8	96.5
25118	7.0031	67.1	79.3	89.3	96.0	89.4	96.1
31622	8.8164	64.1	76.4	88.8	95.6	88.9	95.7
39810	11.0992	61.5	74.0	88.2	95.1	88.3	95.1

***** STRIP AREA NUMBER 10 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.447	0.464	0.0130	1.00	1.000	0.13

GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1121.	10.962	7.630	1.235

RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.176

EMR	RSCAL	RVEL	ELT	TIT	AR
0.447	1.00	1.0000	0.132	0.0130	3.321

NB	NS
22	54

CDROTOR = 1.3326000E-03
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	75.2	86.3	80.9	86.7	81.9	89.5
5011	1.3973	73.8	84.8	82.1	88.0	82.7	89.7
6309	1.7591	70.9	82.1	82.9	88.8	83.1	89.7
7943	2.2146	68.0	79.0	83.3	89.3	83.4	89.7
10000	2.7880	67.8	79.3	83.4	89.6	83.5	90.0
12589	3.5099	69.3	81.2	84.1	90.1	84.2	90.6
15848	4.4187	66.7	78.7	84.4	90.4	84.4	90.7
19952	5.5628	64.0	76.1	84.5	90.6	84.6	90.7
25118	7.0031	62.2	74.5	84.1	90.3	84.2	90.4
31622	8.8164	59.3	71.5	83.7	90.0	83.7	90.0
39810	11.0992	56.6	68.8	83.1	89.5	83.1	89.5

***** STRIP AREA NUMBER 11 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.443	0.436	0.0120	1.00	1.000	0.12
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1120.	10.126	6.770	1.223
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.096
EMR	RSCAL	RVEL	ELT	TIT	AR
0.443	1.00	1.0000	0.119	0.0120	3.369
		NB	NS		
		22	54		

CDROTOR = 6.9932002E-03
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	72.7	83.5	78.1	83.4	79.2	86.4
5011	1.3973	71.9	83.0	80.0	85.3	80.7	87.3
6309	1.7591	70.0	81.1	81.4	86.7	81.7	87.8
7943	2.2146	67.3	78.9	82.5	88.0	82.7	88.5
10000	2.7880	67.2	78.7	83.1	88.6	83.2	89.0
12589	3.5099	67.4	79.2	83.3	88.9	83.4	89.4
15848	4.4187	64.3	76.3	83.2	89.0	83.3	89.2
19952	5.5628	61.5	73.4	82.9	88.8	82.9	88.9
25118	7.0031	59.6	71.6	82.3	88.3	82.3	88.4
31622	8.8164	56.8	68.8	81.7	87.7	81.7	87.8
39810	11.0992	53.9	66.1	80.9	87.0	80.9	87.1

***** STRIP AREA NUMBER 12 WWND = 1.0000

EMA	EMTIP	TI	SINCD	CONTR	L/SSTD
0.438	0.412	0.0240	1.00	1.000	0.11
GAM	RHO	C	SDIA	SPERC	TPR
1.400	0.0790	1119.	9.440	6.770	1.214
RPM	NB	NBSTD	HTR	DTIP	CHDR
9782.2	22	22	0.430	22.000	3.028
EMR	RSCAL	RVEL	ELT	TIT	AR
0.438	1.00	1.0000	0.106	0.0240	3.413
		NB	NS		
		22	54		

CDROTOR = 1.3410000E-02
INLET LENGTH/TIP DIAMETER = 0.9900000

STREAMLINE LIFT COEFFICIENT CALCULATED U SING SCLOPT= 2

*** COMPONENT PWL CONTRIBUTIONS - DB RE 10**(-13) WATTS ***

ONE THIRD OCTAVE

FREQUENCY		DIPOLE		QUADRUPOLE		TOTAL	
HERTZ	F/BPF	INLET	EXHAUST	INLET	EXHAUST	INLET	EXHAUST
3981	1.1099	77.1	87.7	83.0	87.9	84.0	90.8
5011	1.3973	76.4	87.3	85.1	90.1	85.7	91.9
6309	1.7591	74.6	85.5	86.7	91.7	87.0	92.7
7943	2.2146	72.1	83.3	88.1	93.3	88.2	93.7
10000	2.7880	72.2	83.7	88.9	94.2	89.0	94.5
12589	3.5099	72.5	84.0	89.2	94.6	89.3	95.0
15848	4.4187	69.4	81.1	89.2	94.8	89.3	95.0
19952	5.5628	66.8	78.6	89.0	94.7	89.1	94.8
25118	7.0031	64.8	76.6	88.4	94.2	88.4	94.3
31622	8.8164	62.1	74.0	87.8	93.7	87.9	93.7
39810	11.0992	59.3	71.3	87.2	93.1	87.2	93.1

FAN TOTAL POWER SPECTRUM

TOBN	F/BPF	PWL-UP	PWL-DN	PWL-TOT
36	1.1099	103.48	114.46	114.79
37	1.3973	105.39	116.50	116.83
38	1.7591	106.24	117.13	117.47
39	2.2146	106.75	116.87	117.28
40	2.7880	107.50	117.53	117.94
41	3.5099	107.81	117.75	118.17
42	4.4187	107.71	117.61	118.03
43	5.5628	107.24	117.07	117.50
44	7.0031	106.27	115.93	116.38
45	8.8164	105.40	114.80	115.27
46	11.0992	104.82	113.94	114.44

2.2.3.7.2 Directivity Output File from STATOR

STATOR SPL PLOT OUTPUT FILE

FREQUENCY = 3981, OBN = 36

ANGLE INL SPL EXH SPL TOT SPL

10.0	36.9	39.9	41.7
20.0	44.9	50.9	51.9
30.0	56.3	58.8	60.7
40.0	60.1	65.0	66.2
50.0	64.9	70.1	71.3
60.0	70.5	74.4	75.9
70.0	73.7	78.0	79.4
80.0	69.1	83.1	83.2
90.0	58.4	81.3	81.3
100.0	52.1	79.2	79.2
110.0	47.3	77.4	77.4
120.0	42.2	72.9	72.9
130.0	36.5	71.2	71.2
140.0	29.9	72.7	72.7
150.0	21.7	51.3	51.3
160.0	20.2	20.4	23.3
170.0	20.2	20.4	23.3

FREQUENCY = 5011, OBN = 37

ANGLE INL SPL EXH SPL TOT SPL

10.0	36.7	42.2	43.3
20.0	44.0	53.3	53.7
30.0	57.1	61.1	62.6
40.0	60.4	67.4	68.1
50.0	65.8	72.6	73.4
60.0	72.2	76.9	78.1
70.0	76.4	80.5	81.9
80.0	70.1	85.6	85.8
90.0	57.9	83.0	83.0
100.0	52.6	81.1	81.1
110.0	47.9	79.6	79.6
120.0	42.9	74.4	74.4
130.0	37.4	69.8	69.8
140.0	30.9	74.0	74.0
150.0	22.8	50.1	50.1
160.0	20.4	20.6	23.5
170.0	20.4	20.6	23.5

FREQUENCY = 6309, OBN = 38

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		35.3	42.5	43.2
20.0		45.8	53.6	54.2
30.0		58.1	61.4	63.1
40.0		59.7	67.7	68.3
50.0		67.9	73.0	74.1
60.0		73.3	77.3	78.8
70.0		77.5	81.0	82.6
80.0		62.4	86.1	86.1
90.0		65.8	83.8	83.9
100.0		60.4	80.9	80.9
110.0		55.3	81.4	81.4
120.0		50.0	75.1	75.1
130.0		44.3	68.3	68.3
140.0		37.6	75.1	75.1
150.0		29.4	50.1	50.1
160.0		20.2	20.6	23.4
170.0		20.2	20.6	23.4

FREQUENCY = 7943, OBN = 39

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		33.7	42.1	42.7
20.0		45.4	53.2	53.8
30.0		59.1	61.1	63.2
40.0		60.9	67.4	68.2
50.0		67.6	72.6	73.8
60.0		74.0	77.0	78.8
70.0		77.5	80.7	82.4
80.0		70.9	85.8	85.9
90.0		63.4	82.8	82.8
100.0		57.9	80.1	80.1
110.0		52.9	82.0	82.0
120.0		47.8	76.2	76.2
130.0		42.3	66.8	66.8
140.0		35.8	76.4	76.4
150.0		27.8	48.3	48.3
160.0		19.2	20.7	23.0
170.0		19.2	20.7	23.0

FREQUENCY = 10000, OBN = 40

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		30.2	43.4	43.6
20.0		40.7	54.5	54.7
30.0		59.2	62.4	64.1
40.0		61.8	68.7	69.5
50.0		68.3	73.9	75.0
60.0		74.9	78.3	80.0
70.0		78.6	82.0	83.6
80.0		70.1	86.9	87.0
90.0		62.3	82.8	82.8
100.0		57.0	79.8	79.8
110.0		52.3	81.2	81.2
120.0		47.5	77.3	77.3
130.0		42.1	65.3	65.3
140.0		35.8	78.0	78.0
150.0		27.9	50.9	51.0
160.0		19.0	20.9	23.1
170.0		19.0	20.9	23.1

FREQUENCY = 12589, OBN = 41

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		33.2	43.5	43.9
20.0		41.5	54.6	54.8
30.0		58.4	62.5	63.9
40.0		61.0	68.8	69.5
50.0		68.9	74.1	75.2
60.0		75.7	78.5	80.3
70.0		78.9	82.2	83.8
80.0		68.7	86.6	86.7
90.0		61.0	83.4	83.4
100.0		56.0	80.9	80.9
110.0		51.6	80.6	80.6
120.0		47.0	77.0	77.0
130.0		41.8	68.9	68.9
140.0		35.7	79.8	79.8
150.0		27.8	45.2	45.3
160.0		18.7	20.8	22.9
170.0		18.7	20.8	22.9

FREQUENCY = 15848, OBN = 42

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		33.0	43.3	43.7
20.0		40.2	54.4	54.6
30.0		57.3	62.3	63.5
40.0		61.5	68.6	69.4
50.0		69.1	73.9	75.2
60.0		76.4	78.4	80.5
70.0		78.4	82.1	83.6
80.0		66.8	85.7	85.8
90.0		59.5	83.0	83.0
100.0		55.0	81.0	81.0
110.0		50.9	80.4	80.4
120.0		46.5	76.9	76.9
130.0		41.5	68.5	68.6
140.0		35.4	81.5	81.5
150.0		27.7	44.7	44.8
160.0		18.4	20.8	22.8
170.0		18.4	20.8	22.8

FREQUENCY = 19952, OBN = 43

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0		29.6	42.4	42.6
20.0		40.5	53.5	53.7
30.0		53.3	61.4	62.0
40.0		61.5	67.8	68.7
50.0		69.6	73.0	74.7
60.0		76.8	77.5	80.2
70.0		77.2	81.2	82.7
80.0		64.6	84.3	84.4
90.0		58.0	82.3	82.3
100.0		53.9	80.0	80.1
110.0		50.1	80.5	80.5
120.0		45.9	76.8	76.8
130.0		40.9	67.7	67.7
140.0		35.0	82.4	82.4
150.0		27.3	42.9	43.1
160.0		18.0	20.5	22.4
170.0		18.0	20.5	22.4

FREQUENCY = 25118, OBN = 44

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0	26.7	40.7	40.8	
20.0	38.2	51.8	52.0	
30.0	46.7	59.7	59.9	
40.0	60.4	66.0	67.1	
50.0	69.6	71.3	73.5	
60.0	76.7	75.8	79.3	
70.0	74.9	79.5	80.8	
80.0	62.5	82.6	82.7	
90.0	56.5	80.8	80.8	
100.0	52.8	79.0	79.1	
110.0	49.1	79.5	79.5	
120.0	45.0	75.7	75.7	
130.0	40.2	67.4	67.4	
140.0	34.3	82.3	82.3	
150.0	26.7	39.2	39.5	
160.0	17.8	19.9	22.0	
170.0	17.8	19.9	22.0	

FREQUENCY = 31622, OBN = 45

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0	24.2	39.0	39.1	
20.0	37.1	50.1	50.3	
30.0	50.5	58.0	58.7	
40.0	59.0	64.4	65.5	
50.0	68.9	69.6	72.3	
60.0	76.9	74.1	78.7	
70.0	71.6	77.8	78.8	
80.0	60.9	80.9	81.0	
90.0	55.6	78.9	78.9	
100.0	52.2	77.2	77.2	
110.0	48.7	77.9	77.9	
120.0	44.7	74.6	74.6	
130.0	39.9	65.0	65.0	
140.0	34.1	82.3	82.3	
150.0	26.5	35.8	36.3	
160.0	17.8	19.2	21.5	
170.0	17.8	19.2	21.5	

FREQUENCY = 39810, OBN = 46

ANGLE	INL	SPL EXH	SPL TOT	SPL
10.0	24.3	37.9	38.1	
20.0	33.4	49.1	49.2	
30.0	51.9	57.0	58.1	
40.0	57.3	63.3	64.3	
50.0	68.4	68.6	71.5	
60.0	77.1	73.0	78.5	
70.0	65.2	76.8	77.1	
80.0	59.9	79.9	79.9	
90.0	55.3	77.4	77.5	
100.0	52.1	75.4	75.5	
110.0	48.7	76.2	76.2	
120.0	44.8	73.1	73.1	
130.0	40.0	63.0	63.0	
140.0	34.2	82.2	82.2	
150.0	26.7	35.6	36.2	
160.0	17.8	18.5	21.2	
170.0	17.8	18.5	21.2	

2.2.4 BBNPLOTS – Postprocessor

Program BBNPLOTS, a suggested postprocessor implementation, reads output files from programs ROTOR and STATOR and produces plots to help the user visualize and compare the outputs from these programs. The program reads sound power level and sound pressure level (directivity) predictions produced by programs ROTOR and STATOR, calculates logsums of rotor and stator predictions to get totals, and produces input plot files for Unix utility XMGR. The user is presented with a menu of plot types to choose from.

PWL type plots (sound power level vs. frequency) may be made for inlet, exhaust, or total values. Lines will be plotted for rotor, stator, and rotor + stator or for rotor + stator with optional line of input data points. SPL type plots may be sound pressure level versus frequency at a given angle or sound pressure level versus angle at a given frequency. Lines will be plotted for rotor, stator, and rotor + stator or for rotor + stator. SPL type plots can include an optional line of input data points.

BBNPLOTS supplies default titles, labels, and axis limits for the plots it creates, but the user may customize these plot attributes by supplying a plot-options file to override the default plot attributes.

The plot-options file is Fortran namelist format so that only the desired attributes must be supplied. The plot options file is also used to supply data points to be plotted against the ROTOR and STATOR noise predictions.

This implementation uses Unix utility XMGR to produce plots. (See website: <http://plasma-gate.weizmann.ac.il/Xmgr/> to obtain software.)

The program prompts for 4 input files, then presents a menu of 17 options for choosing plot types. An output method is then requested (XMGR plot input files with or without the associated hard copy plots). After that an optional namelist plot options file may be specified and then the program creates the indicated files and/or plots.

2.2.4.1 Description of Input Files

BBNPLOTS requires four input files: the ROTOR power output file, ROTOR directivity output file, STATOR power output file, and STATOR directivity output file. These files are described in Section 2.2.2.4, page 6.

An optional plot-options file (Fortran namelist format) may be used to override program defaults for titles, labels, and axis limits and to supply data points that may be plotted along with the predictions. There is a namelist group corresponding to most plot types. This file may be used if a single plot is to be produced or if all of the PWL type plots are to be produced. The options file will be ignored when multiple SPL plots are to be produced because, so far, there is no namelist group of common plot titles, legend labels, etc. for every possible frequency or angle.

2.2.4.2 Description of Output Files

BBNPLOTS creates an output file, for each plot containing data points and plot attributes, in the format required by plot utility XMGR.

3.0 Fan MPT Superposition Code Descriptions and Users Guides

3.1 MPT Prediction System

To predict MPT spectra, the user needs to:

1. Obtain a CFD code capable of predicting turbomachinery flows in multiple passages. A commercial code or one of the many codes developed by NASA would be suitable.
2. Obtain a grid generator capable of producing meshes for single-passage fan geometries. Again, a commercial code or one of the many grid generators developed by NASA would be suitable.
3. Develop the API routines described in Section 3.3 to generate the multiple-passage meshes required for the prediction.
4. Develop capability to obtain circumferential pressure distributions at the required axial location. (The freely available Visual3 software is recommended.)

MPT predictions can then be made by first generating a single-passage mesh and CFD solution at the required operating point. A multiple-passage mesh is then generated using the ROTBLD program; one blade will have been modified to reflect the geometry change of interest. Circumferential pressure distributions are extracted from these two solutions and input to the SUPERPOSE program together with details of the distribution of geometry changes around the annulus. The output from this program is the predicted MPT spectra.

3.2 Software Installation

The SUPERPOSE and ROTBLD programs are supplied in a gzipped tar file (**superpose.tar.gz**). After decompressing and untaring, the following directories and files should exist:

superpose	Source code and Makefile for SUPERPOSE program
rotbld	Source code and Makefile for ROTBLD program
lib	Source code and Makefile for libraries
inc	Include files
test	Test case
makefile	Makefile for program
make_defs	Compiler options

To make the program type: **make ARCH=*******

In the main directory, the architecture variable is one of:

iris4d	SGI
sgi-irix6	SGI
hp9000s700	HP (PA2 chip)
hp9000s700PA1	HP (PA1 chip)
sun4	Sun
OSF1	Dec
rs6000	IBM

Note GNU make is required for this to work. When the make has finished, the executable resides in **src/ARCH/superpose**. It is possible to make a version for the PC; however the DEC FORTRAN compiler is needed to handle the dynamic allocation.

3.3 ROTBLD Users Guide

This program is designed to interface with a variety of file formats. The file read and write is done through a well-defined API. The skeleton routines for this API are stored in **file_read_api.f**. It is the user's responsibility to modify this to read and write files in their formats.

On executing the program, it will first prompt for the input files; the number of nodes and the geometrical stretching factor for the inlet block are prompted for next. This is the region of mesh added onto the inlet to prevent spurious reflections from contaminating the solution.

The number of blades to be analyzed is now prompted for. For each blade, the geometry may be modified through stagger, thickness, theta shift, or camber changes:

1. **Stagger:** Angle change (degrees) requested, positive is clockwise looking radially inward.
2. **Thickness:** The maximum thickness (units = length) change is first requested. Three scale factors are input to define how this thickness change varies around the blade. The factors are specified at the leading edge, midchord, and trailing edge. Linear variation as assumed between these points.
3. **Theta shift:** The shift in θ (degrees) is requested. This shift is applied uniformly up the span.
4. **Camber:** The maximum camber change (units = length) is now input. As with the thickness variation, three scale factors are input representing the scaled change at the leading edge, midchord, and trailing edge.

3.4 SUPERPOSE Users Guide

On executing the program, it will prompt for three input file names and one output file name. The input files are all ASCII and in free format, as follows:

1. **.mps** file – This is the file giving the circumferential pressure distribution for the multiple passage solution at the axial location of interest. The format is as follows:

```
npass nmps del
x1 p1
x2 p2
.
.
xnmps pnmps
```

where *npass* is the number of passages in the multiple passage solution, *n* is the number of points in the file, and *del* is the geometry change applied to on the blade; *x* and *p* are the circumferential distributions of position and pressure.

2. **.sps** file – This is the file giving the circumferential pressure distribution for a single passage at the axial location of interest. The format is as follows:

```
nsps
x1 p1
x2, p2
.
.xnsps pnsps
```

where n is the number of points in the file; x and p are the circumferential distribution of position and pressure.

3. **psv** file – This file contains the distribution of blade shape variation (del) that the estimate of the MPT spectra is required for:

$nbls$
 del_1
 del_2
 \cdot
 \cdot
 del_{nbls}

4. **.mde** This is the output file containing the predicted amplitudes of the Fourier components:

$1 \ A_m^1 \ A^1 \ B^1$
 $2 \ A_m^2 \ A^2 \ B^2$
 \cdot
 \cdot
 $n \ A_m^n \ A^n \ B^n$

A_m is the magnitude of the fourier mode; (A_i, B_i) are the Fourier series coefficients.

Example input and output files are provided in the test directory. The input files can be in any unit system. The flow chart on page 3 of this document illustrates the flow of data within the MPT predictions system.

4.0 Core Noise Code Description and Users Guide

The following section describes software written to support Contract NAS3-27720, Sub AOI 13.4, Core Noise Model.

4.1 Software Requirements

The core noise program is written in standard Fortran 77.

4.2 Description of Core Noise Code

Initially all inputs needed to run the program are described. Program inputs are read-in through a input data file and output is also written to a designated output file. A first set of calculations to set program constants, calculate frequently derived quantities, etc., is performed. All input is next written to the output file.

The PWL for all frequencies as well as the distribution of this PWL into the the NCOF cut-off ratio bins is initialized to zero. Next, we invoke the principal subroutine STRIP for all the NRHT strips. Prior to this call, a call to INTSTR sets up inputs needed to run STRIP including free-vortex based interpolation of required mean flow properties of the various turbine blade rows based on hub tip ratios fore and aft of each blade row and the nondimensional radial height of each strip. This subroutine returns the PWL contribution from each strip in SWATTIN (for all NF 1/3-octave bands of interest whose center values are input in TOBF) and further in SWCOF the partitioning of SWATTIN into the NCOF cut-off ratio bins (for each of the NF frequencies) is calculated. We store in WATTIN and WATTSCOF the sums of SWATTIN and SWCOF obtained from each strip. Finally E.J. Rice's subroutine BBRDCFCR is called to supply farfield SPL spectra based on WATTSCOF (and other parameters). The Rice module returns SPL spectra for angles in the farfield from DELANG to (180.-DELANG) degrees to the inlet axis and returns two sets SPLTL and SPL which correspond to including a nozzle transmission loss or not including this. As a final step, since the SPL information is required at angles ASPL, the angles calculated by the Rice module that are closest to each of the ASPL are found and the SPL spectra (with and without transmission loss) at these angles are printed out.

All subroutine and function subprograms are comprehensively described within the source code. A description is offered of the key subroutine STRIP.

Inputs to this subroutine are very similar to those of the main program except they are localized to the strip of interest. A key initial check is performed to determine if any blade row has supersonic exit relative Mach number (corresponding to whether the blade row is choked) and (if so) if there is only one blade row which is choked that is also the last blade row. For each frequency, based on conditions in that strip at turbine exit, a range of tangential wave numbers corresponding to waves above cut-off and also discrete values of tangential wave number that yield an integer number of waves that can be accommodated in a circumference are determined. For each such discrete tangential wave number, actuator disk theory is exercised in CALC or CALC1 (depending on the choking issue) to determine the emitted downstream sound wave amplitude (ADSW) corresponding to a unit input "hot spot" wave. Via TSPEC1, TSPEC2 (and also ISPEC, LOVR, LOVR2, etc.) the amplitude of the relevant spectral component of the "hot spot" spectrum is determined. The output is in WATTIN (PWL contributions neglecting exit nozzle transmission loss for all NF 1/3-octave bands)

and in WATTSCOF resides the allocation of WATTIN into the NCOF cut-off ratio bins (some of which may have no power allocated to them).

To avoid taking logarithms of zero or negative quantities, if the program output yields decibel values of -1000 dB for either power levels or SPL's, such values should really be interpreted as vanishingly low values.

4.3 Description of Core Noise Input File

NR denotes the total number of blade rows and $NS=NR+1$ the number of spaces. Thus space 1 denotes "upstream" and space NS denotes "downstream." For axial components, positive is downstream. For tangential (y) components, a direction must be selected as positive and consistently adhered to.

NR =	Number of blade rows .ie. 49 (integer input) such that the last blade row is choked i.e. has subsonic relative mean flow at inlet and supersonic relative mean flow at exit. (note that $ns = nr + 1$)
NF =	Number of 1/3-octave power levels to be calculated
NRHT =	Number of radial heights at which temperature fluctuation and entropy wave length scale inputs are given
NSPL =	Number of far field acoustic angles from inlet axis at which third octave spl spectra (re: $2.10^{**}(-5)$ N/m**2) are desired
TOBF =	Center frequencies of the third octave bands to be calculated, "NF" values in hertz (all positive)
RHT =	NRHT values of nondimensional radial height i.e. $(radius - inner\ radius)/(outer\ radius - inner\ radius)$. Thus: "0.00 .ie. RHT .ie. 1.00"
TFPV =	Percent temperature fluctuations (rms in percent of mean) — NRHT values
LOVRV =	Axial length scale of temperature fluctuations nondimensionalized by radius in upstream annulus — NRHT values (radius corresponds to RHT(I))
LOVR2V =	Tangential length scale of temperature fluctuations nondimensionalized by radius in upstream annulus; NRHT (radius corresponds to RHT(I))
CE =	Downstream static speed of sound: ft/s at the pitch line
GAM =	Specific heat ratio of gas
GAMA =	Specific heat ratio of gas in ambient
PE =	Downstream static pressure, psia, at the pitch line
RMX =	Vector of NS values of axial mach numbers in NS spaces at the pitch line
RMY =	Vector of NS values of tangential mach numbers in NS spaces. Again sign convention must be adhered to at the pitch line.

RPM =	Vector of NR values of wheel rpm's of the blade rows. Consistent sign convention must be adhered to depending on the direction of wheel motion . Value of 0.0 denotes a stator.
DIN =	Vector of NR values of mean diameters upstream of the blade rows (inches) at the pitch line
DOUT =	Vector of NR values of mean diameters downstream of the blade rows (inches) at the pitch line
SIGIN =	Vector of NR values of hub to tip ratios upstream of the blade rows
SIGOUT =	Vector of NR values of hub to tip ratios downstream of the blade rows
CRD =	Vector of (N – 1 = NR) values of ratio static speed of sound in space to static speed of sound in discharge space: CRD(NS) =1.00 (by definition, obviously) at the pitch line.
AXV =	Vector of (NR – 1) values of ratio of spacing between blade row actuator disks to mean radius. AXV(1) and AXV(NR + 1 = NS) are taken as zero. No input for this vector is needed if NR = 1 at the pitch line.
TTOT =	Total temperature in aft duct (°R)
PTOT =	Total pressure in aft duct (psia)
TSUR =	Total temperature in surrounding medium (°R)
PSUR =	Total pressure in surrounding medium, (psia)
HTRAT =	Aft duct hub/tip ratio
ANOZRAT =	Nozzle throat area ÷ aft duct area
DISTANCE =	Radius of microphone array or traverse (ft)
ISIDELN =	0 for constant radius, = 1 for constant sideline calculation
DDUCT =	Aft duct outside diameter, (inches)
FMACH2 =	Mach number of surrounding medium
NCOF =	Number of cut-off ratio bins to sort acoustic power (must be .LE. 200)
DELANG =	Angle increment desired on farfield radial traverse (degrees)
ASPL =	Angles (degrees) from inlet axis at which spl spectra are desired — NSPL values
ISPEC =	Input “1” means correlation function in tangential or axial direction is as $\exp(-r/l)$, “2” means as $\exp(-(r/l)**2)$; “3” means as $1/(1+(r*pi/2l)**2)$

4.4 Description of Core Noise Output File

All the input values are printed out first. The PWL spectrum with and without exit nozzle transmission loss is the first output. Following this , the SPL spectra at the required NF frequencies at angles from the Rice subroutine closest to the required ASPL angles are tabulated. As with the PWL, the SPL values are printed out both with and without exit nozzle transmission loss.

4.5 Core Noise Program Source Code Listing

```
C *****
C *
C *          'CNOISE' PROGRAM
C *
C *****
C
C          THIS PROGRAM CALCULATES THE NOISE GENERATED WHEN
C          "TEMPERATURE FLUCTUATIONS" OF GIVEN AMPLITUDES AND LENGTH SCALES
C          IMPINGE ON A MULTI STAGE TURBINE FROM THE UPSTREAM END. STANDARD ACTUATOR
C          DISK THEORY APPLICABLE TO BLADE ROWS CHARACTERIZED BY SUBSONIC OR SUPERSONIC
C          EXIT RELATIVE MEAN FLOW THROUGH THE BLADE ROW IS USED. UPTO 49 ROWS ARE
C          ALLOWED FOR. THE TURBINE MEAN FLOW DATA IS ASSUMED GIVEN AT PITCH LINE
C          AND IS THEN EXTRAPOLATED TO OTHER RADII BY ASSUMING "FREE VORTEX"
C          DISTRIBUTIONS. THE 'SPL' SPECTRUM AT THE 'NF' FREQUENCIES AT THE 'NSPL'
C          ANGLES IS CALCULATED AS WELL AS THE 'PWL' SPECTRUM.
C
C *****
C
C          INPUT :
C
C          'NR' denotes the total number of blade rows and 'NS=NR+1' the
C          number of spaces . Thus space 1 denotes 'upstream' and space
C          NS denotes 'downstream' . For axial components , positive
C          is downstream . For tangential ( y ) components , a direction
C          must be selected as positive and consistently adhered to.
C
C          NR = NUMBER OF BLADE ROWS .LE. 49 ( integer input )
C          SUCH THAT THE LAST BLADE ROW IS CHOKED i.e. HAS
C          SUBSONIC RELATIVE MEAN FLOW AT INLET AND SUPERSONIC
C          RELATIVE MEAN FLOW AT EXIT.
C
C          NOTE THAT NS = NR + 1
C
C          NF = NUMBER OF THIRD OCTAVE POWER LEVELS TO BE CALCULATED
C
C          NRHT = NUMBER OF RADIAL HEIGHTS AT WHICH TEMPERATURE FLUCTUATION
C          AND ENTROPY WAVE LENGTH SCALE INPUTS ARE GIVEN
C
C          NSPL = NUMBER OF FAR FIELD ACOUSTIC ANGLES FROM INLET AXIS
C          AT WHICH THIRD OCTAVE SPL SPECTRA (RE: 2.10**(-5) N/m**2
C          ARE DESIRED
C
C          TOBF = CENTER FREQUENCIES OF THE THIRD OCTAVE BANDS TO BE
C          CALCULATED, "NF" VALUES IN HERTZ (ALL POSITIVE)
C
C          RHT = 'NRHT' VALUES OF NON DIMENSIONAL RADIAL HEIGHT I.E.
C          (RADIUS-INNER RADIUS)/(OUTER RADIUS-INNER RADIUS).
C          THUS '0.00.LE.RHT.LE.1.00'
C
C          TFPV = PER CENT TEMPERATURE FLUCTUATIONS (RMS IN PERCENT OF
C          MEAN)-'NRHT' VALUES
C
C          LOVRV = AXIAL LENGTH SCALE OF TEMPERATURE FLUCTUATIONS
C          NONDIMENSIONALIZED BY RADIUS IN UPSTREAM ANNULUS-
C          'NRHT' VALUES ( RADIUS CORRESPONDS TO 'RHT(I)' )
C
C          LOVR2V = TANGENTIAL LENGTH SCALE OF TEMPERATURE FLUCTUATIONS
C          NONDIMENSIONALIZED BY RADIUS IN UPSTREAM ANNULUS-
C          'NRHT' VALUES ( RADIUS CORRESPONDS TO 'RHT(I)' )
```

```

C
C      CE = DOWNSTREAM STATIC SPEED OF SOUND : feet/second:
C          AT THE PITCH LINE
C
C      GAM = SPECIFIC HEAT RATIO OF GAS
C
C      GAMA = SPECIFIC HEAT RATIO OF GAS IN AMBIENT
C
C      PE = DOWNSTREAM STATIC PRESSURE, psia, AT THE PITCH LINE
C
C      RMX = VECTOR OF NS VALUES OF AXIAL MACH NUMBERS IN
C          NS SPACES AT THE PITCH LINE
C
C      RMY = VECTOR OF NS VALUES OF TANGENTIAL MACH NUMBERS IN
C          NS SPACES . AGAIN SIGN CONVENTION MUST BE ADHERED
C          TO . AT THE PITCH LINE.
C
C      RPM = VECTOR OF NR VALUES OF WHEEL RPM'S OF THE
C          BLADE ROWS . CONSISTENT SIGN CONVENTION
C          MUST BE ADHERED TO DEPENDING ON THE DIRECTION OF
C          WHEEL MOTION . VALUE OF 0.00 DENOTES A STATOR .
C
C      DIN = VECTOR OF NR VALUES OF MEAN DIAMETERS UPSTREAM
C          OF THE BLADE ROWS ( inches ) AT THE PITCH LINE
C
C      DOUT = VECTOR OF NR VALUES OF MEAN DIAMETERS DOWNSTREAM
C          OF THE BLADE ROWS ( inches ) AT THE PITCH LINE
C
C      SIGIN = VECTOR OF NR VALUES OF HUB TO TIP RATIOS UPSTREAM
C          OF THE BLADE ROWS
C
C      SIGOUT = VECTOR OF NR VALUES OF HUB TO TIP RATIOS DOWNSTREAM
C          OF THE BLADE ROWS
C
C      CRD = VECTOR OF (NS-1=NR) VALUES OF RATIO STATIC SPEED OF
C          SOUND IN SPACE TO STATIC SPEED OF SOUND IN DISCHARGE
C          SPACE : CRD ( NS ) =1.00 ( by definition , obviously ).
C          AT THE PITCH LINE.
C
C      AXV = VECTOR OF (NR-1) VALUES OF RATIO OF SPACING
C          BETWEEN BLADE ROW ACTUATOR DISKS TO MEAN RADIUS.
C          AXV(1) AND AXV(NR+1=NS) ARE TAKEN AS ZERO .
C          NO INPUT FOR THIS VECTOR IS NEEDED IF NR=1 .
C          AT THE PITCH LINE.
C
C      TTOT = TOTAL TEMPERATURE IN AFT DUCT, (DEGREES RANKINE)
C
C      PTOT = TOTAL PRESSURE IN AFT DUCT, (PSIA)
C
C      TSUR = TOTAL TEMPERATURE IN SURROUNDING MEDIUM, (DEGREES RANKINE)
C
C      PSUR = TOTAL PRESURE IN SURROUNDING MEDIUM, (PSIA)
C
C      HTRAT = AFT DUCT HUB-TIP RATIO
C
C      ANOZRAT = (NOZZLE THROAT AREA)/(AFT DUCT AREA)
C
C      DISTANCE = RADIUS OF MICROPHONE ARRAY OR TRAVERSE, (FT.)
C
C      ISIDELN = 0 FOR CONSTANT RADIUS, = 1 FOR CONSTANT SIDELINE CALCULATION

```

```

C
C      DDUCT = AFT DUCT OUTSIDE DIAMETER, (INCHES)
C
C      FMACH2 = MACH NUMBER OF SURROUNDING MEDIUM
C
C      NCOF   = NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
C              (must be .LE. 200)
C
C      DELANG = ANGLE INCREMENT DESIRED ON FAR-FIELD RADIAL TRAVERSE, (DEGREES)
C
C      ASPL   = ANGLES (DEGREES) FROM INLET AXIS AT WHICH SPL SPECTRA
C              ARE DESIRED - 'NSPL' VALUES
C
C      ISPEC   = INPUT "1" MEANS CORRELATION FUNCTION IN TANGENTIAL
C              OR AXIAL DIRECTION IS AS  $\exp(-r/l)$ : "2" MEANS
C              AS  $\exp(-(r/l)**2)$ : "3" MEANS AS  $1/(1+(r*\pi/2l)**2)$ 
C
C *****
C
C      OUTPUT :
C
C              AFT RADIATED THIRD OCTAVE PWL's re:  $10^{(-13)}$  WATTS.
C
C              THIRD OCTAVE SPL RE:  $(2.*10^{(-5)})$  N/m**2)
C
C              RESULTS ARE GIVEN BOTH WITH & WITHOUT TRANS LOSS DUE TO
C
C              EXIT NOZZLE
C
C *****
C
C              BEGIN MAIN PROGRAM
C
C *****
C
C      DIMENSION RPM(50),TOBF(20),PWL(20),TFPV(25),
C      &          WATTSCOF(200,20),ANGLE(200),SPL(200),SPLTL(200),
C      &          SPLFL(200,20),SPLTLFL(200,20),PWLTL(20),DWCOF(200),
C      &          SWCOF(200,20),SWATTIN(20),WATTIN(20),ASPL(20)
C
C      REAL      MXD,MYD,MRD,LOVR,LOVR2,LOVRV(25),LOVR2V(25)
C
C      COMMON    CE,PE,RMX(50),RMY(50),DIN(50),DOUT(50),CRD(50),GAM,
C      &          AXV(50),RHT(25),SIGIN(50),SIGOUT(50),NR,NS,NRHT,
C      &          SMX(50),SMY(50),SDIN(50),SDOUT(50),SCRD(50),
C      &          SAXV(50),ShOVH,SANNHT,SCE,SPE
C
C      CHARACTER*32 INPF
C      CHARACTER*32 OUTF
C
C      C*** READ INPUT AND OUTPUT FILE NAMES ( MUST BE LE. 32 CHARACTERS
C      C*** LONG )
C
C      PRINT *, ' INPUT FILE NAME ? (MUST BE LE. 32 CHARS LONG)'
C      READ 1000, INPF
C      OPEN ( UNIT=7,FILE=INPF,STATUS='OLD' )
C      PRINT *, ' OUTPUT FILE NAME ? (MUST BE LE. 32 CHARS LONG)'
C      READ 1000, OUTF
C      OPEN ( UNIT=8,FILE=OUTF,STATUS='NEW' )
C

```

```

C***  END READ OF INPUT AND OUTPUT FILE NAMES
C
C***  READ IN ALL INPUTS-ALSO SET 'NS' AS 'NR+1'
C
      READ (7,*) NR,NF,NRHT,NSPL
          NS = NR+1
      DO I = 1,NF
          READ (7,*) TOBF(I)
      END DO
      READ (7,*) CE,GAM,GAMA,PE
      DO I = 1,NRHT
          READ (7,*) RHT(I),TFPV(I),LOVRV(I),LOVR2V(I)
      END DO
      READ (7,*) TTOT,PTOT,TSURS,PSURS,HTRAT,ANOZRAT,DISTANCE
      READ (7,*) ISIDELN,DDUCT,FMACH2,NCOF,DELANG
      DO I = 1,NSPL
          READ (7,*) ASPL(I)
      END DO
      DO 10 I = 1,NS
          READ (7,*) RMX(I),RMY(I)
10  CONTINUE
      DO 20 I = 1,NR
          READ (7,*) RPM(I),DIN(I),DOUT(I),CRD(I),SIGIN(I),SIGOUT(I)
20  CONTINUE
      IF ( NR.GE.2 ) THEN
          DO 30 I = 2,NR
              READ (7,*) AXV(I)
30  CONTINUE
      ENDIF
      READ (7,*) NCOF
      READ (7,*) ISPEC
C
C***  END OF READING OF INPUTS
C
C***  PROGRAM CONSTANTS AND SOME RELABELLING,PREPROCESSING OF
C***  INPUTS
C
      AXV(1)   = 0.00
      AXV(NS)  = 0.00
      CRD(NS)  = 1.00
      FCOF     = FLOAT(NCOF)
C***  DISCHARGE RELATED
      MXD      = RMX(NS)
      MYD      = RMY(NS)
      FMACHD   = MXD
      MRD      = SQRT (MXD**2+MYD**2)
      TDUCT    = TTOT/(1.+5*(GAM-1.)*MRD**2)
      CDUCT    = 49.0421*SQRT(TDUCT)
      GC1      = (GAMA-1.)/2.
      GC2      = GAMA/(GAMA-1.)
      TSUR     = TSURS*(1.+GC1*FMACH2**2)
      PSUR     = PSURS*(1.+GC1*FMACH2**2)**GC2
C
C***  END OF CALCULATION OF
C***  PROGRAM CONSTANTS AND SOME RELABELLING,PREPROCESSING OF
C***  INPUTS
C
C***  WRITE OUT INPUT DATA
C
      WRITE (8,1010)

```

```

        WRITE (8,1020) NR,CE,GAM,GAMA,PE
        WRITE (8,1021) TTOT,PTOT,TSURS,PSURS,HTRAT,ANOZRAT
        WRITE (8,1022) DDUCT,FMACH2
        WRITE (8,1025)
        WRITE (8,1026)
        DO I = 1,NRHT
            WRITE (8,1027) RHT(I),TFPV(I),LOVRV(I),LOVR2V(I)
        END DO
C
        DO 70 I = 1,NS
            IF ( I.EQ.1 ) WRITE (8,1030)
            IF ( (I.GT.1).AND.(I.LT.NS) ) WRITE (8,1040) I
            IF ( I.EQ.NS ) WRITE (8,1050)
            WRITE (8,1060) RMX(I),RMY(I)
            IF ( I.NE.NS ) WRITE (8,1070) RPM(I),CRD(I),DIN(I),
&                                DOUT(I),SIGIN(I),SIGOUT(I)
            IF ( (I.GT.1).AND.(I.LT.NS) ) WRITE (8,1080) AXV(I)
        70 CONTINUE
C
        WRITE (8,1090)
C
C***  END OF WRITE OUT OF INPUT DATA
C
C
C***  HEADER FOR WRITE OF OUTPUT DATA
        WRITE (8,1100)
        WRITE (8,1104) ISPEC,NCOF
        WRITE (8,1105)
        WRITE (8,1106)
C
        INITIALIZE 'WATTIN' AND 'WATTSCOF' TO ZERO
        DO I = 1,NF
            WATTIN(I) = 0.00
            DO J = 1,NCOF
                WATTSCOF(J,I) = 0.00
            END DO
        END DO
C
C***  DO VARIOUS STRIPS
C
        DO ISTR = 1,NRHT
            CALL INTSTR ( ISTR )
            TFP = TFPV(ISTR)
            LOVR = LOVRV(ISTR)
            LOVR2 = LOVR2V(ISTR)
            CALL STRIP ( NR,NF,TOBF,TFP,LOVR,LOVR2,SCE,GAM,SPE,
&                        SANNHT,SMX,SMY,RPM,SDIN,SDOUT,SCRD,SAXV,
&                        SHOVH,NCOF,ISPEC,SWATTIN,SWCOF )

            DO I = 1,NF
                WATTIN(I) = WATTIN(I)+SWATTIN(I)
                DO J = 1,NCOF
                    WATTSCOF(J,I) = WATTSCOF(J,I)+SWCOF(J,I)
                END DO
            END DO
        END DO
C
C***  DO VARIOUS FREQUENCY BANDS
C
        DO I = 1,NF
            IF ( WATTIN(I).GT.0.00 ) THEN
                ETAD = TOBF(I)*DDUCT/(12.*CDUCT)

```

```

        PWL(I) = 130.+10.*ALOG10(WATTIN(I))
        DO J = 1,NCOF
            DWCOF(J) = WATTSCOF(J,I)
        END DO
        CALL BBRDCFCR(TTOT,PTOT,TSUR,PSUR,HTRAT,ANOZRAT,
&            DISTANCE,ISIDELN,DDUCT,DJET,FMACHD,FMACH1,FMACH2,
&            NCOF,DWCOF,DELANG,ETAD,NANGLE,ANGLE,SPL,
&            SPLTL,WATTSUM,WATTRAN,FMACHN,COFMIN,CRAT)
        DO IANG = 1,NANGLE
            SPLF(IANG,I) = SPL(IANG)
            SPLTLF(IANG,I) = SPLTL(IANG)
        END DO
        PWLTL(I) = 130.+10.*ALOG10(WATTRAN)
    ELSE
        PWL(I) = -1000.
        DO IANG = 1,NANGLE
            SPL(IANG) = -1000.
            SPLTL(IANG) = -1000.
            SPLF(IANG,I) = SPL(IANG)
            SPLTLF(IANG,I) = SPLTL(IANG)
        END DO
        PWLTL(I) = -1000.
    ENDIF
END DO
C
DO I = 1,NF
    WRITE (8,1107) TOBF(I),PWL(I),PWLTL(I)
END DO
C
C*** WRITE OUT SPL'S AT THE REQUIRED ANGLES
C
DO I = 1,NSPL
    ANGSP = ASPL(I)
    CALL ANGSR ( NANGLE,ANGLE,ANGSP,IANGL,ANGO )
    IF ( ISIDELN.EQ.0 ) WRITE (8,1110) DISTANCE
    IF ( ISIDELN.EQ.1 ) WRITE (8,1115) DISTANCE
    WRITE (8,1104) ISPEC,NCOF
    WRITE (8,1120) ANGO
    WRITE (8,1121)
    DO II = 1,NF
        WRITE (8,1108) TOBF(II),SPLF(IANG,II),SPLTLF(IANG,II)
    END DO
END DO
C
C*** FORMAT STATEMENTS***-----
C
1000 FORMAT(A32)
1010 FORMAT(20X,'***** INPUT PARAMETERS *****',
1         //)
1020 FORMAT(20X,'NUMBER OF BLADE ROWS           =',I8,/,
1         20X,'EXIT STATIC SPD. OF SOUND ( fps ) =',F8.1,/,
2         20X,'SPECIFIC HEAT RATIO OF GAS        =',F8.2,/,
2         20X,'SPECIFIC HEAT RATIO OF GAS-AMBIENT =',F8.2,/,
3         20X,'EXIT STATIC PRESSURE (psia)       =',F8.2,/)
1021 FORMAT(20X,'TOTAL TEMP. IN AFT DUCT,DEG R   =',F8.1,/,
1         20X,'TOTAL PRESSURE IN AFT DUCT ,psia  =',F8.2,/,
2         20X,'AMBIENT STATIC TEMPERATURE,DEG R =',F8.1,/,
3         20X,'AMBIENT STATIC PRESSURE,psia     =',F8.2,/,
4         20X,'AFT DUCT HUB TIP RATIO            =',F8.3,/,
5         20X,'NOZZLE THROAT/AFT DUCT AREA RATIO =',F8.3)

```

```

1022 FORMAT(20X,'AFT DUCT DIAMETER,inches           =',F8.2,/,
1      20X,'AMBIENT MACH NUMBER                     =',F8.3,///)
1025 FORMAT(20X,'TEMPERATURE FLUCTUATION RELATED INPUTS',/)
1026 FORMAT(20X,3X,'RHT  ',2X,'Trms/T %',2X,'Ax L/R',3X,'Ta L/R',
&      //)
1027 FORMAT(20X,1X,F6.2,4X,F6.2,2X,F7.4,2X,F7.4)
1030 FORMAT('1',///,20X,' UPSTREAM SPACE-@ PITCH LINE',/)
1040 FORMAT(///,20X,' SPACE NUMBER      ',I5,' -@ PITCH LINE',/)
1050 FORMAT(///,20X,' DOWNSTREAM SPACE-@ PITCH LINE',/)
1060 FORMAT(20X,'AXIAL MACH NUMBER IN SPACE          =',F8.4,/,
1      20X,'TANGENTIAL MACH NUMBER IN SPACE        =',F8.4)
1070 FORMAT(20X,'WHEEL RPM OF BLADE ROW AT AFT END  ',/,
1      20X,'OF SPACE                                =',F8.1,/,
2      20X,'SPEED OF SOUND IN SPACE NORMALISED BY',/,
3      20X,'EXIT SPEED OF SOUND                    =',F8.4,/,
4      20X,'DIAMETER AHEAD OF BLADE ROW (ins)      =',F8.2,/,
5      20X,'DIAMETER AFT OF BLADE ROW (ins)        =',F8.2,/,
6      20X,'HUB TO TIP RATIO AHEAD OF BLADE ROW    =',F8.4,/,
7      20X,'HUB TO TIP RATIO AFT OF BLADE ROW      =',F8.4,/)
1080 FORMAT(20X,'AXIAL SPACING OF SPACE NORMALISED BY',/,
1      20X,'MEAN RADIUS OF ANNULUS                  =',F8.4)
1090 FORMAT(////,17X,'***END WRITE OF INPUT PARAMETERS***')
1100 FORMAT('1',20X,'*****OUTPUT FROM PROGRAM*****',
1      //)
1104 FORMAT(10X,'EXPONENT RELATED TO CORRELATION FUNCTION = ',
1      I3,/,
2      10X,'# OF CUT OFF RATIO BINS USED IN CALCULATION = ',
3      I3,/)
1105 FORMAT(10X,'THIRD OCTAVE BAND POWER RE: 10**(-13) WATTS',/,
&      10X,'      TL denotes Transmission Loss',/)
1106 FORMAT('      CENTER FREQUENCY,HZ      no TL dB',
&      ' with TL dB',/)
1107 FORMAT('      ',F7.1,'      ',F8.1,2X,F8.1)
1108 FORMAT(10X,'      ',F7.1,'      ',F8.1,2X,F8.1)
1110 FORMAT('1',10X,'MIKES ON AN ARC OF RADIUS ',F8.1,' FT.',/)
1115 FORMAT('1',10X,'MIKES ON A SIDELINE,DISTANCE ',F8.1,' FT.',
&      //)
1120 FORMAT(10X,'THIRD OCTAVE BAND SPL RE: 2*10**(-5) N/m**2',
&      //,10X,'      TL denotes Transmission Loss',/,
&      10X,'      ANGLE FROM INLET (degrees) = ',F8.2,/)
1121 FORMAT(10X,'      CENTER FREQUENCY,HZ      no TL dB',
&      ' with TL dB',/)
C
C***  END OF FORMAT STATEMENTS***-----
C
C      STOP
C
C      END
C
C      *****
C*****  END OF MAIN PROGRAM *****
C      *****
C
C      *****
C*****  SUBROUTINE 'STRIP'*****
C      *****
C
C      THIS SUBROUTINE CALCULATES THE NOISE GENERATED WHEN
C      "TEMPERATURE FLUCTUATIONS" OF A GIVEN AMPLITUDE AND LENGTH SCALE
C      IMPINGE ON A MULTI STAGE TURBINE FROM THE UPSTREAM END.STANDARD ACTUATOR

```

C DISK THEORY APPLICABLE TO BLADE ROWS CHARACTERIZED BY SUBSONIC OR SUPERSONIC
C EXIT RELATIVE MEAN FLOW THROUGH THE BLADE ROW IS USED.UPTO 49 ROWS ARE
C ALLOWED FOR.

C *****

C INPUT :

C Since the subroutine is for an "unwrapped" annular strip ,
C all inputs below are for the relevant strip . 'NR' denotes
C the total number of blade rows and 'NS=NR+1' the number
C of spaces . Thus space 1 denotes 'upstream' and space
C NS denotes ' downstream' . For axial components , positive
C is downstream . For tangential (y) components , a direction
C must be selected as positive and consistently adhered to.

C NR = NUMBER OF BLADE ROWS .LE. 49 (integer input)

C NOTE THAT NS = NR + 1

C NF = NUMBER OF THIRD OCTAVE POWER LEVELS TO BE CALCULATED

C TOBF = CENTER FREQUENCIES OF THE THIRD OCTAVE BANDS TO BE
C CALCULATED,"NF" VALUES IN HERTZ (ALL POSITIVE)

C TFP = PER CENT TEMPERATURE FLUCTUATIONS (RMS IN PERCENT OF
C MEAN)

C LOVR = AXIAL LENGTH SCALE OF TEMPERATURE FLUCTUATIONS
C NONDIMENSIONALIZED BY MEAN RADIUS IN UPSTREAM ANNULUS

C LOVR2 = TANGENTIAL LENGTH SCALE OF TEMPERATURE FLUCTUATIONS
C NONDIMENSIONALIZED BY MEAN RADIUS IN UPSTREAM ANNULUS

C CE = DOWNSTREAM STATIC SPEED OF SOUND : feet/second

C GAM = SPECIFIC HEAT RATIO OF GAS

C PE = DOWNSTREAM STATIC PRESSURE, psia

C ANNHT = RADIAL HEIGHT OF ANNULUS AT EXIT OF LAST BLADE ROW,
C inches

C RMX = VECTOR OF NS VALUES OF AXIAL MACH NUMBERS IN
C NS SPACES .

C RMY = VECTOR OF NS VALUES OF TANGENTIAL MACH NUMBERS IN
C NS SPACES . AGAIN SIGN CONVENTION MUST BE ADHERED
C TO .

C RPM = VECTOR OF NR VALUES OF WHEEL RPM'S OF THE
C BLADE ROWS . CONSISTENT SIGN CONVENTION
C MUST BE ADHERED TO DEPENDING ON THE DIRECTION OF
C WHEEL MOTION . VALUE OF 0.00 DENOTES A STATOR .

C DIN = VECTOR OF NR VALUES OF STRIP MEAN DIAMETERS UPSTREAM
C OF THE BLADE ROWS (inches)

C DOUT = VECTOR OF NR VALUES OF STRIP MEAN DIAMETERS DOWNSTREAM
C OF THE BLADE ROWS (inches)

```

C          CRD = VECTOR OF (NS-1=NR) VALUES OF RATIO STATIC SPEED OF
C          SOUND IN SPACE TO STATIC SPEED OF SOUND IN DISCHARGE
C          SPACE : CRD ( NS ) =1.00 ( by definition , obviously )
C
C          AXV = VECTOR OF (NR-1) VALUES OF RATIO OF SPACING
C          BETWEEN BLADE ROW ACTUATOR DISKS TO MEAN RADIUS.
C          AXV(1) AND AXV(NR+1=NS) ARE TAKEN AS ZERO .
C          NO INPUT FOR THIS VECTOR IS NEEDED IF NR=1 .
C
C          HOVH = ANNULAR STRIP HEIGHT/ANNULUS HEIGHT
C
C          NCOF  = NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
C          (must be .LE. 200)
C
C          ISPEC  = INPUT "1" MEANS CORRELATION FUNCTION IN TANGENTIAL
C          OR AXIAL DIRECTION IS AS EXP(-r/l):"2" MEANS
C          AS EXP (-(r/l)**2):"3" MEANS AS 1/(1+(r*pi/2l)**2)
C
C          NON DIMENSIONALISATION OF WAVE AMPLITUDES:
C
C          SOUND WAVES : If "p'" is amplitude of fluctuating pressure,non
C          dimensional amplitude is "p'/(gam*Pst)" where "Pst"
C          is the mean static pressure in the relevant space.
C
C          SHEAR WAVE  : If "v'" is the amplitude of fluctuating tangential
C          velocity, non dimensional amplitude is "v'/C" where
C          "C" is the mean speed of sound in the relevant space.
C
C          ENTROPY WAVE: If "s'" is the amplitude of fluctuating entropy,then
C          non dimensional amplitude is "s'/Cp" where "Cp" is
C          the specific heat at constant pressure.
C
C          *****
C
C          OUTPUT :
C
C          1.AFT RADIATED THIRD OCTAVE POWER IN WATTS FOR 'NF' FREQUENCIES,
C          AS 'WATTIN(I),I=1,NF'
C          2.AFT RADIATED THIRD OCTAVE POWER IN WATTS FOR 'NF' FREQUENCIES
C          SEGREGATED INTO 'NCOF' CUT-OFF RATIO BINS,AS 'WATTSCOF(JR,I),
C          JR=1,NCOF AND I=1,NF'.
C
C          *****
C
C          SOME COMMENTS CONCERNING THE PRESENT SUBROUTINE:
C
C          1. PROPAGATION EFFECTS DUE TO THE PRESCENCE OF ADJACENT
C          BLADE ROWS ARE INCLUDED IN THE ACTUATOR DISK
C          APPROXIMATION VALID FOR BLADE ROWS CARRYING SUBSONIC
C          EXIT RELATIVE MEAN FLOW i.e ASSUMING CONTINUITY OF
C          MASS FLUX,STAGNATION ENTHALPY RELATIVE TO THE BLADE ROW
C          AND NO CHANGE IN FLOW ANGLE AT THE TRAILING EDGE.FOR CHOKED
C          BLADE ROWS, A CHOKING CONDITION PREVAILS DOWNSTREAM OF THE
C          CHOKED ROW AND IS USED IN PLACE OF THE "NO CHANGE IN FLOW
C          ANGLE" CONDITION AT THE TRAILING EDGE.
C          2. ONLY AN ENTROPY WAVE IS PRESUMED INCIDENT ON THE UPSTREAM
C          BLADE ROW.
C          3. THE TWO DIMENSIONAL ACTUATOR DISK APPROACH IS BASED ON
C          LINEARISING ABOUT UNIFORM FLOWS ( DIFFERENT ON
C          EITHER SIDE OF A BLADE ROW ).

```

```

C      4. REFLECTION FREE CONDITIONS ARE ASSUMED BEYOND LAST ROW.
C
C *****
C
C      SUBROUTINE STRIP ( NR,NF,TOBF,TFP,LOVR,LOVR2,CE,GAM,PE,
&                      ANNHT,RMX,RMY,RPM,DIN,DOUT,CRD,AXV,
&                      HOVH,NCOF,ISPEC,WATTIN,WATTSCOF )
C
C      DIMENSION RMT(50),RMX(50),RMY(50),RPM(50),CRD(50),AXV(50),
&              TOBF(20),RAD(50),DIN(50),DOUT(50),ICH(50),
&              WATTSCOF(200,20),WATTIN(20)
C
C      REAL      KIN(50),KY,MXD,MYD,MRD,MRU,LOVR,KU,KETA,KXI,
&              KIN1,KINR,LOVR2,MT,MYR,MX,MREL
C
C      COMPLEX    CZERO,IM,CUNITY
C
C*** PROGRAM CONSTANTS AND SOME RELABELLING,PREPROCESSING OF
C*** INPUTS
C
C      NS        = NR+1
C      AXV(1)    = 0.00
C      AXV(NS)   = 0.00
C      CRD(NS)   = 1.00
C      FCOF      = FLOAT(NCOF)
C      PI        = 3.141593
C*** SWITCH "RPM'S" TO WHEEL TIP MACH NUMBERS BASED ON 'CE'
C      FACT      = PI/(1440.*CE)
C      DO 60 I = 1,NR
C          RMT(I) = FACT*RPM(I)*(DIN(I)+DOUT(I))
C          RAD(I) = (DIN(I)+DOUT(I))/48.00
C      60 CONTINUE
C*** AREA OF DISCHARGE ANNULUS IN SQUARE FEET
C      AREA      = PI*(DIN(NR)+DOUT(NR))*ANNHT/288.
C*** "CZERO" DENOTES THE COMPLEX NUMBER WHOSE REAL AND IMAGINARY PARTS
C*** ARE BOTH ZERO
C      CZERO     = CMPLX(0.00,0.00)
C*** "CUNITY" DENOTES THE COMPLEX NUMBER WHOSE REAL PART IS UNITY AND
C*** IMAGINARY PART IS ZERO
C      CUNITY    = CMPLX(1.00,0.00)
C*** DETERMINE BLADE ROWS WITH SUPERSONIC EXIT RELATIVE MACH NUMBERS
C*** I.E. WHICH ARE CHOKED
C      NCHR      = 0
C      DO I = 1,NR
C          ICH(I) = 0
C      END DO
C      DO I = 2,NS
C          IM1    = I-1
C          MT     = RMT(IM1)/CRD(I)
C          MX     = RMX(I)
C          MYR    = RMY(I)-MT
C          MREL   = SQRT(MX**2+MYR**2)
C          IF ( MREL.GT.1. ) NCHR = NCHR+1
C          IF ( MREL.GT.1. ) ICH(NCHR) = IM1
C      END DO
C*** 'ICHCR' IS SET 'GT' 1 IF A NON ZERO NUMBER OF ROWS IS CHOKED AND
C*** IT IS NOT THE CASE THAT ONLY THE LAST BLADE ROW IS CHOKED.
C      ICHCR     = 0
C      IF ( (NCHR.GT.0).AND.(ICH(1).LT.NR) ) ICHCR = 1
C*** DISCHARGE & UPSTREAM RELATED

```

```

MRU      = SQRT(RMX(1)**2+RMY(1)**2)
CTHU     = RMX(1)/MRU
TTHU     = RMY(1)/RMX(1)
MXD      = RMX(NS)
MYD      = RMY(NS)
MRD      = SQRT (MXD**2+MYD**2)
OMMRD2   = 1.-MRD**2
SRMRD2   = SQRT(OMMRD2)
SRMXD2   = SQRT(1.-MXD**2)
DPHI     = PI/100.
CON      = (0.2316/MRU)*(GAM*PE*144.*CE)*1.3558*AREA*
&        HOVH*LOVR*LOVR2*(TFP/100.)**2*2./PI**2
C***  "IM" DENOTES THE COMPLEX NUMBER WHOSE REAL PART IS ZERO AND
C***  IMAGINARY PART IS UNITY
IM       = CMPLX(0.00,1.00)
C
C***  END OF CALCULATION OF
C***  PROGRAM CONSTANTS AND SOME RELABELLING,PREPROCESSING OF
C***  INPUTS
C
C***  DO VARIOUS FREQUENCY BANDS
C
DO I = 1,NF
  DO J = 1,NR
    KIN(J) = 2.*PI*TOBF(I)*RAD(J)/CE
  END DO
  DO JR = 1,NCOF
    WATTSCOF(JR,I)=0.00
  ENDDO
  KINR    = KIN(NR)
  KIN1    = KIN(1)
  KU      = KIN1/CRD(1)
  KXI     = KU/MRU
  IF ( ISPEC.EQ.1 ) THEN
    TSPEC1 = 1./(1.+(KXI*LOVR)**2)
  ELSE
    IF ( ISPEC.EQ.2 ) THEN
      TSPEC1 = EXP(-(KXI*LOVR)**2/PI)
    ELSE
      TSPEC1 = EXP(-2.*KXI*LOVR/PI)
    END IF
  END IF
  JMIN    = -KINR*MYD/OMMRD2-KINR*SRMRD2/OMMRD2
  JMAX    = -KINR*MYD/OMMRD2+KINR*SRMRD2/OMMRD2
  SUMM    = 0.00
C
DO J = JMIN,JMAX
  FJ      = FLOAT(J)
  KY      = FJ
  S       = (KY*OMMRD2+KINR*MYD)/(KINR*SRMRD2)
  C       = SQRT(1.-S**2)
  IF ( ICHCR.EQ.0 ) THEN
    CALL CALC ( NR,KY,KIN,AXV,RMX,RMY,RMT,CRD,GAM,ADSW )
  ELSE
    CALL CALC1 ( NR,KY,KIN,AXV,RMX,RMY,RMT,CRD,GAM,NCHR,
&              ICH,ADSW )
  END IF
  IF ( FJ.GT.0.00 ) CTF = (SRMRD2-MYD)/(S*SRMRD2-MYD)
  IF ( FJ.LT.0.00 ) CTF = (SRMRD2+MYD)/(MYD-S*SRMRD2)
  IF ( FJ.EQ.0.00 ) CTF = 1.E+09

```

```

CTFR2 = CTF**2
TJR1  = (1.-1./CTFR2)
TJR   = FCOF*TJR1
JR    = 1+INT(TJR)
IF ( JR.GT.NCOF ) JR = NCOF
KETA  = KY/CTHU-KU*TTHU/MRU
IF ( ISPEC.EQ.1 ) THEN
    TSPEC2 = 1./(1.+(FJ*LOVR2)**2)
ELSE
    IF ( ISPEC.EQ.2 ) THEN
        TSPEC2 = EXP(-(FJ*LOVR2)**2/PI)
    ELSE
        AFJ     = ABS(FJ)
        TSPEC2 = EXP(-2.*AFJ*LOVR2/PI)
    END IF
END IF
PHIT  = TSPEC1*TSPEC2
AEF   = (OMMRD2*SROMXD2/(SROMXD2-MYD*S-SROMRD2*MXD*C))*
&      (MXD+(C*SROMRD2+MXD*MYD*S-MXD*SROMXD2)/
&      (SROMXD2-MXD*SROMRD2*C-MYD*S))
TSUM  = PHIT*ADSW**2*AEF
SUMM  = SUMM+TSUM
WATTSCOF(JR,I) = WATTSCOF(JR,I)+TSUM*CON*KU
END DO

C
WATTIN(I) = SUMM*CON*KU
C
END DO
C
RETURN
C
END

C
C *****
C ***** END OF SUBROUTINE 'STRIP' *****
C *****
C
C *****
C ***** FUNCTION 'SGNF' *****
C *****
C
C *** FUNCTION SGNF ( I,J ) GIVES (-1)**(I+J) WHICH IS +1 IF (I+J) IS
C *** EVEN AND -1 IF (I+J) IS ODD .
C
FUNCTION SGNF ( I,J )
C
IPJ      = I+J
IDISC    = MOD ( IPJ,2 )
IF ( IDISC.EQ.0 ) SGNF = 1.00
IF ( IDISC.NE.0 ) SGNF = -1.00
C
RETURN
C
END

C
C *****
C ***** END OF FUNCTION 'SGNF' *****
C *****
C
C *****

```

```

C***** FUNCTION DETM *****
C
C
C*** FUNCTION DETM ( D ) YIELDS THE COMPLEX DETERMINANT ' DETM ' OF
C*** THE COMPLEX 2X2 MATRIX ' D '.
C
C      COMPLEX FUNCTION DETM ( D )
C
C      COMPLEX D(2,2)
C      DETM      = D(1,1)*D(2,2)-D(1,2)*D(2,1)
C
C      RETURN
C
C      END
C
C      *****
C***** END OF FUNCTION ' DETM ' *****
C
C      *****
C***** SUBROUTINE MATINV *****
C
C      *****
C*** SUBROUTINE MATINV INVERTS THE COMPLEX 3X3 MATRIX 'INP' STORING
C*** THE RESULT IN 'INV'. THE PROCEDURE INVOLVES COMPUTING THE
C*** COFACTORS 'A' OF 'INP' WHICH ITSELF ENTAILS THE CALCULATION OF
C*** THE MINORS OF 'INP'.
C
C      SUBROUTINE MATINV ( INP,INV )
C
C      COMPLEX INP(3,3),INV(3,3),A(3,3),D(2,2),DET,DETM
C
C      DO 400 I = 1,3
C        DO 300 J = 1,3
C*** SET UP 2X2 MATRIX 'D' FOR EVALUATION OF COFACTOR
C        II      = 0
C        DO 200 I1 = 1,3
C          IF ( I1.NE.I ) THEN
C            II = II+1
C            JJ = 0
C            DO 100 J1 = 1,3
C              IF ( J1.NE.J ) THEN
C                JJ      = JJ+1
C                D(II,JJ) = INP(I1,J1)
C              ENDIF
C            100      CONTINUE
C          200      CONTINUE
C*** EVALUATE COFACTOR
C          A(I,J) = DETM(D)*SGNF(I,J)
C        300      CONTINUE
C      400 CONTINUE
C*** EVALUATE DETERMINANT OF OVERALL MATIX
C      DET = CMPLX( 0.00,0.00 )
C      DO 500 J = 1,3
C        DET = DET+A(1,J)*INP(1,J)
C      500 CONTINUE
C*** EVALUTE THE INVERSE MATRIX
C      DO 700 I = 1,3
C        DO 600 J = 1,3

```

```

        INV(I,J) = A(J,I)/DET
600    CONTINUE
700    CONTINUE
C
    RETURN
C
    END
C
C
C *****
C ***** END OF SUBROUTINE 'MATINV' *****
C *****
C
C *****
C ***** SUBROUTINE MATINV1 *****
C *****
C
C *** SUBROUTINE MATINV1 INVERTS THE COMPLEX 3X4 MATRIX 'INP' STORING
C *** THE RESULT IN 'INV'. THE MATRIX "INP" HAS 3 ROWS AND 4 COLUMNS
C *** AND IS PART OF AN ORIGINAL 4 by 4 MATRIX WHOSE FOURTH ROW IS
C *** (0,0,0,1). "INV" HAS THE SAME STRUCTURE AS "INP."
C
    SUBROUTINE MATINV1 ( INP,INV )
C
    COMPLEX INP(3,4),INV(3,4),A(3,3),AI(3,3),B(3),BO(3)
C
C *** SET UP 3X3 MATRIX 'A' FOR INVERSION
    DO 200 I = 1,3
        DO 100 J = 1,3
            A(I,J) = INP(I,J)
        100    CONTINUE
    200    CONTINUE
C *** INVERT "A" AS "AI"
    CALL MATINV ( A,AI )
C
C *** FOURTH COLUMN OF "INP"
    DO 300 I =1,3
        B(I) = INP(I,4)
    300    CONTINUE
C *** NEGATIVE OF FOURTH COLUMN OF "INV"
    CALL MATPRD ( AI,B,BO )
C *** SET UP 3X3 PART OF "INV"
    DO 500 I = 1,3
        DO 400 J = 1,3
            INV(I,J) = AI(I,J)
        400    CONTINUE
    500    CONTINUE
C *** FOURTH COLUMN OF "INV"
    DO 600 I =1,3
        INV(I,4) = -BO(I)
    600    CONTINUE
C
    RETURN
C
    END
C
C
C *****
C ***** END OF SUBROUTINE 'MATINV1' *****
C *****
C
C *****

```

```

C***** SUBROUTINE 'WNUMB' *****
C
C
C
C   SUBROUTINE WNUMB COMPUTES THE (COMPLEX) AXIAL WAVE NUMBER OF THE
C   DOWNSTREAM SOUND WAVE IN KX(1), OF THE UPSTREAM SOUND WAVE IN KX(2) AND
C   OF THE REAL AXIAL WAVENUMBER OF THE SHEAR WAVE IN KXSH.
C
C   SUBROUTINE WNUMB ( K,KY,MX,MY,KX,KXSH )
C   COMPLEX DELB,IM,KX(2)
C   DIMENSION SGN(2)
C   REAL K,KXSH,KY,MX,MY
C
C   SGN(1)   =  1.00
C   SGN(2)   = -1.00
C   IM       = CMPLX(0.00,1.00)
C   A1       =  1.00
C   SK       = SIGN(A1,K)
C
C   DEL      = K-KY*MY
C   OMMX2    = 1.00-MX**2
C   ARG2     = DEL**2-KY**2*OMMX2
C   AARG     = SQRT(ABS(ARG2))
C   IF ( ARG2.GE.0.00 ) DELB = AARG*SK
C   IF ( ARG2.LT.0.00 ) DELB = IM*AARG
C   DO 100 I = 1,2
C       KX(I) = (-MX*DEL+SGN(I)*DELB)/OMMX2
100 CONTINUE
C   KXSH     = DEL/MX
C
C   RETURN
C   END
C
C
C   *****
C***** END OF SUBROUTINE 'WNUMB' *****
C
C
C   *****
C***** SUBROUTINE 'FLUXES' *****
C
C
C
C   SUBROUTINE FLUXES COMPUTES LINEARISED AXIAL MASS FLUX AND
C   STAGNATION ENTHALPY PER UNIT MASS RELATIVE TO BLADE ROW
C   FOR SOUND AND SHEAR WAVES.
C
C   SUBROUTINE FLUXES ( CR,MT,MX,MY,P,U,V,MAF,SEF )
C   COMPLEX MAF,P,RHO,SEF,U,V
C   REAL MT,MX,MY
C
C   RHO      = P
C   MAF      = RHO+U/MX
C   SEF      = (P+MX*U+(MY-MT)*V)*CR**2
C
C   RETURN
C
C   END
C
C
C   *****
C***** END OF SUBROUTINE 'FLUXES' *****
C
C
C

```

```

C                      *****
C***** SUBROUTINE 'UVP' *****
C                      *****
C
C      SUBROUTINE UVP COMPUTES THE NON DIMENSIONAL AXIAL VELOCITY,
C      TANGENTIAL VELOCITY,PRESSURE ASSOCIATED WITH SOUND WAVES FOR
C      I = 1 or 2 AND WITH SHEAR WAVES FOR I = 3.
C
C      SUBROUTINE UVP ( I,K,KX,KXSH,KY,MX,MY,P,U,V )
C      COMPLEX KX,K0,P,U,V
C      REAL K,KXSH,KY,MX,MY
C
C      IM3      = I-3
C      IF ( IM3.NE.0 ) THEN
C          K0      = K-KX*MX-KY*MY
C          U        = KX/K0
C          V        = KY/K0
C          P        = CMPLX(1.00,0.00)
C      ELSE
C          U        = -KY/KXSH
C          V        = CMPLX(1.00,0.00)
C          P        = CMPLX(0.00,0.00)
C      ENDIF
C
C      RETURN
C
C      END
C
C                      *****
C***** END OF SUBROUTINE 'UVP' *****
C                      *****
C
C                      *****
C***** SUBROUTINE 'MATPRD' *****
C                      *****
C
C      YIELDS PRODUCT OF 3 by 3 MATRIX (A) AND A COLUMN VECTOR (B)
C      AND RESULT IS COLUMN VECTOR (C).
C
C      SUBROUTINE MATPRD ( A,B,C )
C      COMPLEX A(3,3),B(3),C(3)
C
C      DO 200 I = 1,3
C          C(I) = CMPLX(0.00,0.00)
C          DO 100 J = 1,3
C              C(I) = A(I,J)*B(J)+C(I)
C          100 CONTINUE
C      200 CONTINUE
C
C      RETURN
C
C      END
C
C                      *****
C***** END OF SUBROUTINE 'MATPRD' *****
C                      *****
C
C                      *****
C***** SUBROUTINE 'MATPRD1' *****
C                      *****

```

```

C
C      "A" IS A COMPLEX 3 by 4 MATRIX AS IS "AOUT." "B" IS A COMPLEX 2 by 4
C      MATRIX AS IS "BOUT." BOTH "A" AND "B" ARE INPUTS.
C      WE OBTAIN A PRODUCT "AOUT" WHICH IS A 3 BY 4 OBTAINED
C      BY PREMULTPLYING "B" by "A."
C      UNDEFINED ROWS OF "A" (4th ROW) AND OF "B"
C      (3rd AND 4th ROWS) ARE TAKEN AS ZERO FOR PURPOSES OF THE MATRIX
C      PRODUCT EVALUATION EXCEPT THAT THE DIAGONAL 4-4 ELEMENT OF BOTH
C      A,B IS UNITY.
C
C      SUBROUTINE MATPRD1 ( A,B,AOUT )
C      COMPLEX A(3,4),B(2,4),AOUT(3,4)
C
C      INITIALIZE AOUT TO ZERO
C      DO 200 I = 1,3
C        DO 100 J = 1,4
C          AOUT(I,J) = CMPLX(0.00,0.00)
100    CONTINUE
200    CONTINUE
C
C      DO 302 I = 1,3
C        DO 202 J = 1,4
C          AOUT(I,J) = CMPLX(0.00,0.00)
C          DO 102 K = 1,2
C            AOUT(I,J) = AOUT(I,J)+A(I,K)*B(K,J)
102    CONTINUE
C            IF ( J.EQ.4 ) AOUT(I,J) = AOUT(I,J)+A(I,4)
202    CONTINUE
302    CONTINUE
C
C      RETURN
C      END
C
C      *****
C ***** END OF SUBROUTINE 'MATPRD1' *****
C *****
C
C      *****
C ***** SUBROUTINE 'SETMAT' *****
C *****
C
C      THIS SUBROUTINE SETS UP THE MATRICES "BU" (WHOSE THIRD ROW HAS ALL
C      ZEROS AND FOURTH ROW HAS THREE ZEROS FOLLOWED BY UNITY) AND "BD"
C      (WHOSE FOURTH ROW HAS THREE ZEROS FOLLOWED BY UNITY). BECAUSE OF THE
C      PROPERTIES OF "BU" AND "BD" , "BU" IS CALCULATED AS A 2 by 4 MATRIX
C      AND "BD" AS A 3 by 4 MATRIX. THE CONSERVED QUANTITY AND WAVE TYPE
C      CONVENTION FOR BOTH "BU" AND "BD" ARE AS UNDER. "R1,R2,R3" ARE PHASE
C      FACTORS GIVING THE CHANGE OF AMPLITUDE & PHASE OF DOWNSTREAM SOUND,
C      UPSTREAM SOUND, SHEAR & ENTROPY WAVES FROM UPSTREAM TO DOWNSTREAM
C      BLADE ROW.
C
C      ROW (CONSERVED QTY)          COLUMN (WAVE TYPE)
C                                  DNSTR SOUND-UPSTR SOUND-SHEAR-ENTROPY
C      LINEARISED MASS FLUX
C      LINEARISED SGN. ENTH.
C      per unit mass relative
C      to blade row
C      CHOKING OR KUTTA CODITION
C      ENTROPY
C

```

```

C      INTEGER INPUT "IMAT" IS USED TO CALCULATE (i) "BU" ONLY IN WHICH
C      CASE INPUTS "MTUPS,AXSP" AND OUTPUTS "BD,R1,R2,R3" ARE NOT USED
C      -SET "IMAT" EQUAL TO "1" (ii) "BD" ONLY IN WHICH CASE INPUTS
C      "MTDNS,AXSP" AND OUTPUTS "BU,R1,R2,R3" ARE NOT USED-SET "IMAT"
C      EQUAL TO "2" AND (iii) BOTH "BD,BU" IN WHICH CASE ALL INPUTS AND
C      ALL OUTPUTS ARE USED-SET "IMAT" EQUAL TO "3"
C
C      SUBROUTINE SETMAT ( IMAT,CR,G,KIN,KY,MTDNS,MTUPS,
1      MX,MY,AXSP,BD,BU,R1,R2,R3 )
C      COMPLEX BD(3,4),BU(2,4),R1,R2,R3,IM,KX(2),MFL,SE,KXI,APCH
C      COMPLEX R1E,R2E,R3E,U,V,P
C
C      REAL KIN,KY,MTDNS,MTUPS,MX,MY,K,KXSH,MTU,MTD,MRELU
C
C      IU      = 0
C      CG      = CR**2/(G-1.)
C      IM      = CMPLX(0.00,1.00)
C      K       = KIN/CR
C      MTD     = MTDNS/CR
C      MTU     = MTUPS/CR
C      MRELU   = SQRT(MX**2+(MY-MTU)**2)
C      IF ( MRELU.GE.1.00 ) IU = 1
C      IF ( MRELU.GE.1.00 ) TMREL = 1.-MRELU**2
C
C      I = 1 , LINEARIZED AXIAL MASS FLUX
C      I = 2 , LINEARIZED SGN. ENTH./UNIT MASS RELATIVE TO UPSTREAM
C      (IN CASE OF "BD") AND DOWNSTREAM (IN CASE OF "BU") BLADE
C      ROWS
C      I = 3 , CHOKING OR KUTTA CONDITION RELATIVE TO UPSTREAM BLADE ROW
C      (FOR "BD")
C      J = 1 , DOWNSTREAM SOUND
C      J = 2 , UPSTREAM SOUND
C      J = 3 , SHEAR
C      J = 4 , ENTROPY
C
C      CALL WNUMB ( K,KY,MX,MY,KX,KXSH )
C      CALCULATE "BD"
C      IF ( (IMAT.EQ.2).OR.(IMAT.EQ.3) ) THEN
C          KXI      = KX(1)
C          CALL UVP ( 1,K,KXI,KXSH,KY,MX,MY,P,U,V )
C          CALL FLUXES ( CR,MTU,MX,MY,P,U,V,MFL,SE )
C          IF ( IU.EQ.0 ) CALL ANGPF ( MTU,MX,MY,U,V,APCH )
C          IF ( IU.EQ.1 ) CALL CHOKP ( G,MTU,MX,MY,U,V,P,APCH )
C          BD(1,1) = MFL
C          BD(2,1) = SE
C          BD(3,1) = APCH
C
C          KXI      = KX(2)
C          CALL UVP ( 2,K,KXI,KXSH,KY,MX,MY,P,U,V )
C          CALL FLUXES ( CR,MTU,MX,MY,P,U,V,MFL,SE )
C          IF ( IU.EQ.0 ) CALL ANGPF ( MTU,MX,MY,U,V,APCH )
C          IF ( IU.EQ.1 ) CALL CHOKP ( G,MTU,MX,MY,U,V,P,APCH )
C          BD(1,2) = MFL
C          BD(2,2) = SE
C          BD(3,2) = APCH
C
C          KXI      = CMPLX(0.00,0.00)
C          CALL UVP ( 3,K,KXI,KXSH,KY,MX,MY,P,U,V )
C          CALL FLUXES ( CR,MTU,MX,MY,P,U,V,MFL,SE )
C          IF ( IU.EQ.0 ) CALL ANGPF ( MTU,MX,MY,U,V,APCH )

```

```

        IF ( IU.EQ.1 ) CALL CHOKP ( G,MTU,MX,MY,U,V,P,APCH )
        BD(1,3) = MFL
        BD(2,3) = SE
        BD(3,3) = APCH
C
        BD(1,4) = CMPLX(-1.00,0.00)
        BD(2,4) = CMPLX(CG,0.00)
        IF ( IU.EQ.0 ) BD(3,4) = CMPLX(0.00,0.00)
        IF ( IU.EQ.1 ) BD(3,4) = CMPLX(0.50,0.00)*TMREL
END IF
C
END OF "BD" CALCULATION
C
C
C
CALCULATE "BU"
IF ( (IMAT.EQ.1).OR.(IMAT.EQ.3) ) THEN
    KXI = KX(1)
    CALL UVP ( 1,K,KXI,KXSH,KY,MX,MY,P,U,V )
    CALL FLUXES ( CR,MTD,MX,MY,P,U,V,MFL,SE )
    BU(1,1) = MFL
    BU(2,1) = SE
C
    KXI = KX(2)
    CALL UVP ( 2,K,KXI,KXSH,KY,MX,MY,P,U,V )
    CALL FLUXES ( CR,MTD,MX,MY,P,U,V,MFL,SE )
    BU(1,2) = MFL
    BU(2,2) = SE
C
    KXI = CMPLX(0.00,0.00)
    CALL UVP ( 3,K,KXI,KXSH,KY,MX,MY,P,U,V )
    CALL FLUXES ( CR,MTD,MX,MY,P,U,V,MFL,SE )
    BU(1,3) = MFL
    BU(2,3) = SE
C
    BU(1,4) = CMPLX(-1.00,0.00)
    BU(2,4) = CMPLX(CG,0.00)
END IF
C
END OF "BD" CALCULATION
C
C
C
CALCULATE R1,R2,R3
IF ( IMAT.EQ.3 ) THEN
    R1E = IM*KX(1)*AXSP
    R2E = IM*KX(2)*AXSP
    R3E = IM*KXSH*AXSP
    R3 = CEXP(R3E)
    R2 = CEXP(R2E)
    R1 = CEXP(R1E)
END IF
C
RETURN
END
C
C
C *****
C ***** END OF SUBROUTINE 'SETMAT' *****
C *****
C
C *****
C ***** SUBROUTINE 'ANGP' *****
C *****
C
C
C
C SUBROUTINE ANGP COMPUTES LINEARISED ANGLE PERTURBATION
C RELATIVE TO BLADE ROW.

```

```

C
  SUBROUTINE ANGP ( MT,MX,MY,U,V,AP )
  COMPLEX AP,U,V
  REAL MT,MX,MY
C
  TANA      = (MY-MT)/MX
  C2A       = 1./(1.+TANA**2)
C
  AP        = (V-TANA*U)*C2A/MX
C
  RETURN
C
  END
C
C      *****
C ***** END OF SUBROUTINE 'ANGP' *****
C      *****
C
C      *****
C ***** SUBROUTINE 'CHOKP' *****
C      *****
C
C  SUBROUTINE CHOKP COMPUTES A LINEARISED QUANTITY THE SUM OF WHICH
C  DUE TO ALL WAVE SYSTEMS NEEDS TO BE ZERO DOWNSTREAM OF A CHOKED
C  BLADE ROW FOR SOND AND SHEAR WAVES.
C
C  SUBROUTINE CHOKP ( G,MT,MX,MY,U,V,P,CH )
C  COMPLEX CH,U,V,P,PTRM,WTRM,THTRM
C  REAL MT,MX,MY,M,M2
C
C  M2       = MX**2+(MY-MT)**2
C  M        = SQRT(M2)
C  CTH      = MX/M
C  STH      = (MY-MT)/M
C  TTH      = STH/CTH
C  GCON     = (G-1.)/2.
C  THCON    = (1.+GCON*M2)*TTH
C
C  PTRM     = GCON*P*(1.-M2)
C  WTRM     = -(U*CTH+V*STH)*(1.-M2)/M
C  THTRM    = ((V*CTH-U*STH)/M)*THCON
C
C  CH       = PTRM+WTRM+THTRM
C
  RETURN
C
  END
C
C      *****
C ***** END OF SUBROUTINE 'CHOKP' *****
C      *****
C
C      *****
C ***** SUBROUTINE 'CALC' *****
C      *****
C
C  "CALC" CARRIES OUT CORE CALCULATIONS FOR DETERMINING RESPONSE OF
C  MULTI BLADE ROW TO AN INCIDENT ENTROPY WAVE IN THE CASE OF NO
C  CHOKED ROWS OR ONLY LAST BLADE ROW CHOKED.
C

```

```

      SUBROUTINE CALC ( NR,KY,KIN,AXV,RMX,RMY,RMT,CRD,GAM,ADSW )
      DIMENSION RMT(50),RMX(50),RMY(50),CRD(50),AXV(50)
C
      REAL      KIN(50),KY,MTDNS,MTUPS,MX,MY
C
      COMPLEX   CZERO,BD(3,4),BU(2,4),IM,SU(3,4),TPROD(3,4),
1             CUNITY,BI(3,4),BO(3,4),R1,R2,R3,BDI(3,4),R3CUM,
2             VI(4),VO(4),BOI(3,4),RHS(3),
3             BUP(2,4),SU4,SUT(3,4)
C***  "CZERO" DENOTES THE COMPLEX NUMBER WHOSE REAL AND IMAGINARY PARTS
C***  ARE BOTH ZERO
      CZERO    = CMPLX(0.00,0.00)
C***  "CUNITY" DENOTES THE COMPLEX NUMBER WHOSE REAL PART IS UNITY AND
C***  IMAGINARY PART IS ZERO
      CUNITY   = CMPLX(1.00,0.00)
C***  "IM" DENOTES THE COMPLEX NUMBER WHOSE REAL PART IS ZERO AND
C***  IMAGINARY PART IS UNITY
      IM       = CMPLX(0.00,1.00)
C
      IWAVE    = 4
      NS       = NR+1
C  INITIALIZE "INCOMING" WAVE SYSTEM AND SET "OUTGOING" WAVE SYSTEM
C  TO ZERO
      DO I = 1,4
         IF ( I.NE.IWAVE ) VI(I) = CZERO
         IF ( I.EQ.IWAVE ) VI(I) = CUNITY
         VO(I) = CZERO
      END DO
C  "R3CUM" IS PHASE FACTOR FOR CONVECTED WAVES
      R3CUM    = CUNITY
      IMAT     = 1
      CR       = CRD(1)
      MTDNS    = RMT(1)
      MTUPS    = RMT(1)
      MX       = RMX(1)
      MY       = RMY(1)
      AXSP     = AXV(1)
C  SET "BUP" & "SU"
      CALL SETMAT ( IMAT,CR,GAM,KIN(1),KY,MTDNS,MTUPS,
1                MX,MY,AXSP,BD,BU,R1,R2,R3 )
      DO I = 1,3
         DO J = 1,4
            IF ( NR.EQ.1 ) THEN
               IF ( I.EQ.3 ) SU(I,J) = CZERO
               IF ( I.LT.3 ) SU(I,J) = BU(I,J)
            ELSE
               IF ( I.EQ.J ) SU(I,J) = CUNITY
               IF ( I.NE.J ) SU(I,J) = CZERO
            END IF
            IF ( I.LT.3 ) BUP(I,J) = BU(I,J)
         END DO
      END DO
      SU4      = R3CUM
C  DOWNSTREAM SPACE CALCULATION IF THERE IS ONLY ONE BLADE ROW
C  I.E SET "BD" IF NR=1
      IF ( NR.EQ.1 ) THEN
         IMAT   = 2
         CR     = 1.00
         MTDNS  = RMT(1)
         MTUPS  = RMT(1)

```

```

        MX      = RMX(2)
        MY      = RMY(2)
        AXSP    = AXV(2)
        CALL SETMAT ( IMAT,CR,GAM,KIN(1),KY,MTDNS,MTUPS,
1          MX,MY,AXSP,BD,BU,R1,R2,R3 )
    ENDIF
C  CALCULATION TO DOWNSTREAM SPACE IF THERE IS MORE THAN ONE BLADE ROW
    IF ( NR.GT.1 ) THEN
        DO 110 I = 2,NR
            IM1      = I-1
            IMAT      = 3
            CR        = CRD(I)
            MTDNS     = RMT(I)
            MTUPS     = RMT(IM1)
            MX        = RMX(I)
            MY        = RMY(I)
            AXSP      = AXV(I)
            CALL SETMAT ( IMAT,CR,GAM,KIN(I),KY,MTDNS,MTUPS,
1          MX,MY,AXSP,BD,BU,R1,R2,R3 )
            R3CUM     = R3CUM*R3
            CALL MATINV1 ( BD,BDI )
            CALL MATPRD1 ( BDI,BUP,TPROD )
            CALL MPRD34 ( TPROD,SU,SU4,SUT )
            CALL SHIFT ( SUT,R1,R2,R3,SU )
            SU4       = R3CUM
            DO L = 1,2
                DO M = 1,4
                    BUP(L,M) = BU(L,M)
                END DO
            END DO
110    CONTINUE
C  CALCULATE "BD" FOR DOWNSTREAM SPACE: ALSO FINAL "SU"
        IMAT      = 2
        CR        = 1.00
        MTDNS     = RMT(NR)
        MTUPS     = RMT(NR)
        MX        = RMX(NS)
        MY        = RMY(NS)
        AXSP      = AXV(NS)
        CALL SETMAT ( IMAT,CR,GAM,KIN(NR),KY,MTDNS,MTUPS,
1          MX,MY,AXSP,BD,BU,R1,R2,R3 )
C
        DO L = 1,3
            DO M = 1,4
                SUT ( L,M ) = SU ( L,M )
                IF ( L.NE.3 ) TPROD(L,M) = BUP(L,M)
                IF ( L.EQ.3 ) TPROD(L,M) = CZERO
            END DO
        END DO
        CALL MPRD34 ( TPROD,SUT,SU4,SU )
    ENDIF
C
C  CALCULATE "BI" AND "BO" BASICALLY BY RESETTING COLUMNS IN "SU"
C  AND "BD" AND CHANGING SIGNS.ALSO OBTAIN "VO(4)"EXPLICITLY SINCE
C  INCIDENT WAVE IS AN ENTROPY WAVE
C
        VO(4)     = VI(4)*R3CUM
        DO 210 I = 1,3
            DO 200 J = 1,4
                IF ( J.NE.2 ) THEN

```

```

        IF ( I.NE.3 ) BI(I,J) = SU(I,J)
        IF ( I.EQ.3 ) BI(I,J) = CZERO
        BO(I,J) = BD(I,J)
    ELSE
        BI(I,J) = -BD(I,J)
        IF ( I.NE.3 ) BO(I,J) = -SU(I,J)
        IF ( I.EQ.3 ) BO(I,J) = CZERO
    END IF
200    CONTINUE
210    CONTINUE
    DO 310 I =1,3
        RHS(I) = CZERO
        DO 300 J = 1,4
            RHS(I) = RHS(I)+BI(I,J)*VI(J)
300    CONTINUE
310    CONTINUE
        CALL MATINV1 ( BO,BOI )
        DO 410 I =1,3
            VO(I) = BOI(I,4)*VO(4)
            DO 400 J = 1,3
                VO(I) = VO(I)+BOI(I,J)*RHS(J)
400    CONTINUE
410    CONTINUE
C
        ADSW = CABS (VO(1))
C
        RETURN
    END
C
*****
C***** END OF SUBROUTINE 'CALC' *****
C*****
C
*****
C***** SUBROUTINE 'CALC1' *****
C*****
C
C    "CALC1" CARRIES OUT CORE CALCULATIONS FOR DETERMINING RESPONSE OF
C    MULTI BLADE ROW TO AN INCIDENT ENTROPY WAVE IN THE CASE OF NON ZERO #
C    OF CHOKED ROWS AND WHEN IT IS NOT THE CASE THAT THE ONLY CHOKED ROW
C    IS THE LAST ROW.
C
    SUBROUTINE CALC1 ( NR,KY,KIN,AXV,RMX,RMY,RMT,CRD,GAM,NCHR,ICH,
&                    ADSW )
    DIMENSION RMT(50),RMX(50),RMY(50),CRD(50),AXV(50),ICH(50)
C
    REAL          KIN(50),KY,MTDNS,MTUPS,MX,MY
C
    COMPLEX       CZERO,BD(3,4),BU(2,4),IM,SU(3,4),SEFL,
1               CUNITY,R1,R2,R3,BDI(3,4),R3CUM,MFL,
2               EWAVE,ENT,BC(4),ALP(3),P(4),MFLN,SEFLN,
3               BUP(2,4),SU4,SUT(3,4),SD(3,4),AL(4)
C***  "CZERO" DENOTES THE COMPLEX NUMBER WHOSE REAL AND IMAGINARY PARTS
C***  ARE BOTH ZERO
        CZERO = CMPLX(0.00,0.00)
C***  "CUNITY" DENOTES THE COMPLEX NUMBER WHOSE REAL PART IS UNITY AND
C***  IMAGINARY PART IS ZERO
        CUNITY = CMPLX(1.00,0.00)
C***  "IM" DENOTES THE COMPLEX NUMBER WHOSE REAL PART IS ZERO AND
C***  IMAGINARY PART IS UNITY
        IM = CMPLX(0.00,1.00)

```

```

C      IWAVE      = 4
      NS         = NR+1
      ICHOKE      = 0
C      INITIALIZE WAVE SYSTEM TO LEFT OF BLADE ROW
C      TO ZERO EXCEPT FOR INCOMING ENTROPY WAVE
      DO I = 1,4
        IF ( I.NE.IWAVE ) AL(I) = CZERO
        IF ( I.EQ.IWAVE ) AL(I) = CUNITY
      END DO
C      START WITH UPSTREAM SPACE BOUNDED TO THE RIGHT BY THE FIRST BLADE ROW.
C      "R3CUM" IS PHASE FACTOR FOR CONVECTED WAVES
      IR          = 1
      R3CUM       = CUNITY
      R1          = CUNITY
      R2          = CUNITY
      R3          = CUNITY
      EWAVE       = R3CUM
      IMAT        = 1
      CR          = CRD(1)
      MTDNS       = RMT(1)
      MTUPS       = RMT(1)
      MX          = RMX(1)
      MY          = RMY(1)
      AXSP        = AXV(1)
      CALL SETMAT ( IMAT,CR,GAM,KIN(1),KY,MTDNS,MTUPS,
1      MX,MY,AXSP,BD,BU,R1,R2,R3 )
      CALL CHOKE ( NCHR,IR,ICH,IC )
      ICPR        = IC
      IF ( IC.EQ.1 ) THEN
        ICHOKE    = ICHOKE+1
        CALL SMATCH ( CR,GAM,KIN(1),KY,MTDNS,MX,MY,BC )
        AL(2)     = -BC(4)*AL(4)/BC(2)
        DO IJ = 1,3
          ALP(IJ) = AL(IJ)
        END DO
        CALL MSFL ( ALP,EWAVE,BU,MFL,SEFL )
        ENT       = EWAVE
      ELSE
        CALL EQAT1 ( BU,BUP )
        CALL DIAG ( R1,R2,R3,SU,SU4 )
      END IF
C
C      CALCULATION TO DOWNSTREAM SPACE
C
      DO 110 I = 2,NR
        IM1       = I-1
        IMAT      = 3
        CR        = CRD(I)
        MTDNS     = RMT(I)
        MTUPS     = RMT(IM1)
        MX        = RMX(I)
        MY        = RMY(I)
        AXSP      = AXV(I)
        CALL SETMAT ( IMAT,CR,GAM,KIN(I),KY,MTDNS,MTUPS,
1      MX,MY,AXSP,BD,BU,R1,R2,R3 )
        R3CUM     = R3CUM*R3
        EWAVE     = R3CUM
C
        IF ( ICPR.EQ.1 ) THEN

```

```

        CALL DIAG ( R1,R2,R3,SU,SU4 )
        CALL EQAT ( BD,SD )
ELSE
    CALL MATINV1 ( BD,BDI )
    CALL MPR4 ( BDI,BUP,SU,SU4,R1,R2,R3,SUT )
    CALL EQAT ( SUT,SU )
    SU4      = SU4*R3
END IF
C
    CALL CHOKE ( NCHR,I,ICH,IC )
    ICPR      = IC
    IF ( IC.EQ.1 ) THEN
        ICHOKE = ICHOKE+1
        CALL SMATCH ( CR,GAM,KIN(1),KY,MTDNS,MX,MY,BC )
        IF ( ICHOKE.EQ.1 ) THEN
            DO M = 1,4
                P(M) = CMPLX(0.00,0.00)
                DO L = 1,3
                    P(M) = P(M)+BC(L)*SU(L,M)
                END DO
            END DO
            P(4) = P(4)+BC(4)*SU4
            AL(2) = -P(4)/P(2)
            DO L = 1,3
                ALP(L) = CMPLX(0.00,0.00)
                DO M = 1,4
                    ALP(L) = ALP(L)+SU(L,M)*AL(M)
                END DO
            END DO
            CALL MSFL ( ALP,EWAVE,BU,MFL,SEFL )
        ELSE
            CALL CHSOL ( BC,SU,SU4,SD,BU,ENT,EWAVE,MFL,SEFL,
&                      MFLN,SEFLN )
            MFL = MFLN
            SEFL = SEFLN
        END IF
        ENT = EWAVE
    ELSE
        CALL EQAT1 ( BU,BUP )
    END IF
110  CONTINUE
C    CALCULATE "BD" FOR DOWNSTREAM SPACE AND 'ADSW' I.E. REQUIRED OUTPUT
    IMAT      = 2
    CR         = 1.00
    MTDNS      = RMT(NR)
    MTUPS      = RMT(NR)
    MX         = RMX(NS)
    MY         = RMY(NS)
    AXSP       = AXV(NS)
    CALL SETMAT ( IMAT,CR,GAM,KIN(NR),KY,MTDNS,MTUPS,
1          MX,MY,AXSP,BD,BU,R1,R2,R3 )
C
    IF ( ICPR.EQ.1 ) THEN
        CALL DICH ( BD,MFL,SEFL,ENT,ADSW )
    ELSE
        CALL DISOL ( SU,SD,ENT,MFL,SEFL,ADSW )
    END IF
C
    RETURN
END

```

```

C
C *****
C ***** END OF SUBROUTINE 'CALC1' *****
C *****
C
C
C **** AFT CORE RADIATION SUBROUTINE, E. J. RICE, REVISED 02/19/1998
C **** NORMALIZED PLANE WAVE RADIATION AND NEW TERMINATION TRANSMISSION
C **** LOSS MODEL INCLUDED HERE.
C ** THE FOLLOWING "BBRDCFCR" IS THE MAIN SUBROUTINE. ALL ARGUMENTS
C ARE INPUTS EXCEPT "DJET," "FMACH1," "NANGLE," "ANGLE," "SPL,"
C "SPLTL," "WATTS," "FMACHN," AND "COFMIN" AND WATTS
C RETURNED TO THE MAIN PROGRAM FOR PRINTING, FILE STORAGE ETC.
C
C ** THIS VERSION REMOVES THE P**2 OSCILLATIONS BEYOND THE PEAK. FOUR
C REGIONS (ETA AND CUT-OFF RATIO) DETERMINE PROPER APPROXIMATION FOR
C HIGH ANGLE RADIATION FOR THE NON-PLANE WAVES. DEVELOPED 01/30/98.

SUBROUTINE BBRDCFCR(TTCOR,PTCOR,TSUR,PSUR,HTRATC,ANOZRATC,
1DISTANCE,ISIDELN,DDUCTC,DJET,FMACHC,FMACH1,FMACH2,
2NCOF,WATTSCOF,DELANG,ETAC,NANGLE,ANGLE,SPL,SPLTL,WATTS,
3WATTRAN,FMACHN,COFMIN,CRAT)

C
C DIMENSION ANGLE(200),SPL(200),SPLTL(200),WATTSCOF(200),
1COFRAT(200),COFRATD(200),COFRATN(200),PSQTOT(200),PSQTLOS(200),
2PSQRADT(200)

C ***** 02/19/1998 *****
C
C ** SUBROUTINE REQUIRED "CORNOZ"
C
C *****
C NOTE! SUBROUTINE "CORNOZ" COULD BE MOVED OUT OF THIS SUBROUTINE TO
C SAVE COMPUTATION TIME IF DESIRED. THE NOZZLE FLOW CONDITIONS, USING
C AN ITERATION, ARE CALCULATED EVERY TIME THIS SUBROUTINE IS ENTERED.
C OF COURSE, THE SUBROUTINE ARGUMENTS MUST BE CAREFULLY CONSTRUCTED TO
C GET ALL OF THE ARGUMENTS PROPERLY INCLUDED WHERE THEY ARE NEEDED.
C *****
C
C ***** DEFINITION OF SUBROUTINE ARGUMENTS *****
C
C ANGLE VECTOR OF ANGLES ON THE FAR-FIELD TRAVERSE, CONVERTED TO
C THAT MEASURED FROM THE ENGINE "INLET" AXIS, (DEGREES)
C ANOZRATC (CORE NOZZLE THROAT AREA)/(CORE DUCT AREA)
C COFMIN THE MINIMUM CUT-OFF RATIO BELOW WHICH COMPLETE REFLECTION
C OCCURS DUE TO REFRACTION THROUGH THE SLIP LAYER
C DDUCTC AFT CORE DUCT OUTSIDE DIAMETER, (INCHES)
C DELANG ANGLE INCREMENT DESIRED ON FAR-FIELD RADIAL TRAVERSE, (DEGREES)
C DISTANCE RADIUS OF MICROPHONE ARRAY OR TRAVERSE, (FT.)
C DJET FINAL JET DIAMETER, (INCHES)
C FMACHC AFT CORE DUCT MACH NUMBER, POSITIVE FOR EXHAUST
C FMACHN NOZZLE EXIT (THROAT) MACH NUMBER
C FMACH1 FINAL JET MACH NUMBER
C FMACH2 MACH NUMBER OF SURROUNDING MEDIUM
C FREQ FREQUENCY OF SOUND, (HERTZ)
C HTRATC AFT CORE DUCT HUB-TIP RATIO
C ISIDELN = 0 FOR CONSTANT RADIUS, = 1 FOR CONSTANT SIDELINE CALCULATION
C NANGLE NUMBER OF ANGLES USED IN CALCULATION, DETERMINED BY DELANG AND
C THE MAXIMUM ANGLE OF 180 DEGREES
C NCOF NUMBER OF CUT-OFF RATIO BINS TO SORT ACOUSTIC POWER
C PSUR TOTAL PRESURE IN SURROUNDING MEDIUM, (PSIA)
C PTCOR TOTAL PRESSURE IN AFT FAN DUCT, (PSIA)
C SPL THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE

```

```

C      "IGNORING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
C      2*10**(-5) NEWTONS/METER**2
C SPLTL  THE SOUND PRESSURE LEVEL VECTOR ON THE FAR-FIELD TRAVERSE
C      "INCLUDING" THE EXIT TRANSMISSION LOSS (DECIBELS) RELATIVE TO
C      2*10**(-5) NEWTONS/METER**2
C TSUR   TOTAL TEMPERATURE IN SURROUNDING MEDIUM, (DEGREES RANKINE)
C TTCOR  TOTAL TEMPERATURE IN AFT CORE DUCT, (DEGREES RANKINE)
C WATTRAN SUM OF TRANSMITTED ACOUSTIC POWER IN ALL BINS, (WATTS)
C WATTS  SUM OF ACOUSTIC POWER IN ALL THE CUT-OFF RATIO BINS, (WATTS)
C WATTSCOF VECTOR WITH ACOUSTIC POWER FOR EACH BIN, (WATTS)
C
      PI = 3.1415927
      QAFP = 1.0+0.328766*ETAC**1.702882
      AFPOWFAC = 1.741*(QAFP+1.274989*ETAC**2)/(PI*ETAC*QAFP)

C  CALCULATE COEF. (PLANE WAVE) INTEGRATION OF RAD. DIRECTIVITY, MACH=0
      IF(ETAC.LT.1.0) THEN
        ACOEFPW = 1.733303+5.30259*ETAC**2.28937
        GO TO 14
      END IF
      ACOEFPW = 7.035893*ETAC**1.773669
14  CONTINUE

C
C  CALC. COEF. FOR INTEGRATION WITH AFT SHEAR LAYERS, PLANE WAVE
C
      AFPOWFPW = (1.0+0.127683*ETAC)/(3.0+0.137590*ETAC)
      PSQCOEFP = ACOEFPW*AFPOWFPW*(1.0+2.6557*ETAC)/
&      (1.0515+3.8508*ETAC)
C *****

      FCOF = NCOF
      FCOFINV = 1./FCOF
      FCOFIND2 = 0.5/FCOF
C ***** SET UP CUT-OFF RATIOS IN THE DUCT *****

      COFSQPR = 1.0
      DO 20 I=1,NCOF
        COFRATD(I) = SQRT(1.0/(COFSQPR-FCOFIND2))
        COFSQPR = COFSQPR-FCOFINV
20  CONTINUE

      TDUCT = TTCOR/(1.0+0.2*FMACHC**2)
      CDUCT = 49.0421*SQRT(TDUCT)

      QSUR = 1.0+0.2*FMACH2**2
      TSTS = TSUR/QSUR
      PSTS = PSUR/QSUR**3.5
      CSUR = 49.0421*SQRT(TSTS)
      RHOSUR = 144.0*PSTS/(53.3*TSTS)

C
C ***** DETERMINE NOZZLE FLOW PROPERTIES *****

      CALL CORNOZ(TTCOR,PTCOR,PSTS,HTRATC,ANOZRATC,DDUCTC,FMACHC,
1FMACH1,CJET,DJET,TJET,PNOZ,DNOZ,CNOZ,FMACHN)

      CRAT = CJET/CSUR

      ETA = ETAC*DJET*CDUCT/(DDUCTC*CJET)
      ETAN= ETAC*DNOZ*CDUCT/(DDUCTC*CNOZ)

      RATCFNOZ = DNOZ*CDUCT*SQRT(1.0-FMACHC**2)/(DDUCTC*CNOZ*

```

```

1SQRT(1.0-FMACHN**2))

RATCFJET = DJET*CDUCT*SQRT(1.0-FMACHC**2)/(DDUCTC*CJET*
1SQRT(1.0-FMACH1**2))

FMSQ1 = FMACH1**2
FM11 = 1.0-FMSQ1
BETA1 = SQRT(FM11)
FMSQ2 = FMACH2**2
C
C ***** ESTABLISH CUT-OFF LIMITS FOR COMPLETE REFLECTION
COFMIN = 1.0
IF(FMACH1.EQ.0.0) GO TO 15

CKM2 = 1.0-CRAT*FM11/FMACH1
IF(FMACH2.LT.CKM2) THEN
COSPHIL = -CRAT/(1.0+CRAT*FMACH1-FMACH2)
PHIL = ACOS(COSPHIL)
SINPHIL = SIN(PHIL)
DEN = SQRT(1.0+FMSQ1+2.0*FMACH1*COSPHIL)
COSPSIL = (COSPHIL+FMACH1)/DEN
SINPSIL = SINPHIL/DEN
COFMIN = SQRT(FM11+FMSQ1*COSPSIL**2)/(BETA1*SINPSIL)
C
COFMIN = (1.0+FMACH1*COSPHIL)/(BETA1*SINPHIL)
END IF
15 CONTINUE
C *****
NANGLE = 180.0/DELANG-1
DO 5 I=1,NANGLE
FI = I
ANGLE(I) = FI*DELANG
5 CONTINUE

C ***** PART OF EMPIRICAL COEFFICIENT TAKING PLACE OF RADICAL
C ***** SQRT(1-1/COF**2) IN P**2 COEFFICIENT WHICH WOULD NOT BE
C ***** ANY GOOD AT CUT-OFF. FOR NON-PLANE WAVES.

ACOE = 0.7/ETA

C ** RECALL, THE CUT-OFF RATIOS IN THE DUCT ARE INPUT HERE AS "COFRATD"

C ***** CALCULATE CUT-OFF RATIOS IN THE NOZZLE AND THE JET *****
DO 22 I=1,NCOF
COFRATN(I) = RATCFNOZ*COFRATD(I)
COFRAT(I) = RATCFJET*COFRATD(I)
22 CONTINUE
C
C ***** INITIALIZE P**2 AT EACH FAR-FIELD ANGLE *****
C
DO 10 I=1,NANGLE
PSQRADT(I) = 0.0
PSQTLOS(I) = 0.0
PSQTOT(I) = 0.0
10 CONTINUE
C
C ***** START LOOP ON CUT-OFF RATIO *****
C
POWCON = 8.36424*RHOSUR*CSUR
WATTS = 0.0

```

```

        WATTRAN = 0.0
        DO 70 J=1,NCOF
        WATTS = WATTS+WATTSCOF(J)
C
C ** DETERMINE IF COFRAT IS WITHIN COMPLETE REFLECTION RANGE DUE TO
C ** REFRACTION THROUGH THE SLIP LAYER OR IF COMPLETE REFLECTION
C ** OCCURS AT THE NOZZLE THROAT
C
        IF(COFRAT(J).LT.COFMIN.OR.COFRATN(J).LE.1.0) GO TO 70

C *****
C !!!!!!! VERY IMPORTANT! FOLLOWING DEFINES PLANE WAVE REGIME !!!!!!!
C *****

C ***** DETERMINE IF PLANE WAVE RADIATION ALGORITHM SHOULD BE USED
C ***** IF PLANE WAVE IPW = 1 (YES) FOR COFRATD>=3 OR LAST BIN
        IPW = 0
        IF(COFRATD(J).GE.3.0.OR.J.EQ.NCOF) IPW=1

C ** CALCULATE TRANSMISSION LOSS AT NOZZLE THROAT CUT-OFF RATIO AND
C ** FREQUENCY PARAMETER.

        FMSQN = FMACHN**2

        IF(IPW.EQ.1) THEN

C ***** IN PLANE-WAVE REGIME *****

        X = 0.5*(PI*ETAN)**2
        RADRES = 1.0+X*EXP(-0.325226*X)-EXP(-0.101669*ETAN**5.7848)
        A = 0.023567
        Y = 0.5*PI**2*ETAN
        RADREC=EXP(-3.574331*ETAN**1.957292)*8.*ETAN/3.+A*Y**2/
&      (1.+A*Y**3)
        QDEN = (1.0+FMACHN)**2*((RADRES+1.0)**2+RADREC**2)
        TLCF = 4.*(RADRES*(1.+FMSQN)+FMACHN*
&      (RADRES**2+RADREC**2+1.))/QDEN
        GO TO 55
        END IF

C ***** IN NON-PLANE WAVE, RADIAL OR SPINNING MODE REGION *****

        QF = PI*ETAN*(1.0-1.0/COFRATN(J))
        QF15SQ = (QF-1.5)**2
        RADREC = 1.135*EXP(-0.29*(QF+0.18)**2)

        IF(QF.LE.1.5) THEN
        RADRES = 1.5*EXP(-0.2124*QF15SQ)
        GO TO 53
        END IF

        RADRES = 1.0+0.5*EXP(-0.5338*QF15SQ)
53 CONTINUE

        TAU = SQRT(1.0-1.0/COFRATN(J)**2)
        TPM = TAU+FMACHN
        TTM = TAU*FMACHN
        QDEN = (RADRES*TPM+TTM+1.0)**2+(RADREC*TPM)**2

```

```

      QNUM = (RADRES+FMACHN)*(RADRES*FMACHN+1.0)+FMACHN*RADREC**2
      TLCF = 4.0*TAU*QNUM/QDEN

C ***** CALCULATE TRANSMITTED POWER *****
C
55 CONTINUE
      WATTRAN = WATTRAN+TLCF*WATTSCOF(J)

C ***** CALCULATE P**2 COEF. TO ACCOUNT FOR WATTS *****
C      TO ACCOUNT FOR MIC. RADIUS, DIVIDE BY DISTANCE**2 AT PSQ BELOW.
C
      POWCOEF = POWCON*WATTSCOF(J)

C ***** PROGRAM SPITS HERE. IT WILL CONTINUE ON FOR NON-PLANE
C ***** WAVES AND WILL JUMP FOR THE PLANE WAVE

      IF(IPW.EQ.1) GO TO 45

C ***** NOW IN "NON-PLANE" WAVE REGIME !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

      COFBETIN = 1.0/(COFRAT(J)*BETA1)
      COFINV = 1.0/COFRAT(J)
      COFINVSQ = COFINV**2
      COFM1 = 1.0-COFINVSQ
      EP = SQRT(COFM1)
      GDEN = (1.0+EP)**2

C ***** HERE IS REMAINDER OF EMPIRICAL COEFFICIENT USING "ACOE" ABOVE

      A90 = 2.0*(ACOE+EP)/(ACOE+1.0)

C ***** THEORETICAL NORMALIZATION COEFFICIENT WITH FLOW ATTACHED TO A90
C ** INCLUDES INTEGRATED POWER NORMALIZATION "AFPOWFAC" (EMPIRICAL) **

      PSQCOEF = AFPOWFAC*A90*(1.0-FMSQ1*COFM1)**1.5/BETA1
      COSPK1 = BETA1*EP/SQRT(1.0-FMSQ1*COFM1)
      ANGPK1 = ACOS(COSPK1)

C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 1, PSIPK1 (DEGREES)
      PSIPK1 = ANGPK1*180.0/PI
      SINPK1 = SIN(ANGPK1)

C ***** PHASE VELOCITY VECTOR ANGLE FOR PEAK IN REGION 1 (JET)
      SIN2 = SINPK1**2
      COSPHI1 = -FMACH1*SIN2+COSPK1*SQRT(1.0-FMSQ1*SIN2)

C ***** PHI1 TO PHI2 ACROSS SLIP LAYER
C ***** PHASE VELOCITY ANGLE IN REGION 2 (SURROUNDINGS)

      COSPHI2 = COSPHI1/(CRAT+(CRAT*FMACH1-FMACH2)*COSPHI1)

C ***** PHASE TO GROUP VELOCITY ANGLES IN REGION 2 (SURROUNDINGS)

      COSPSI2 = (COSPHI2+FMACH2)/SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
      PSI2RAD = ACOS(COSPSI2)
      SINPSI2 = SIN(PSI2RAD)

C ***** ANGLE CHANGE ACOUSTIC POWER CORRECTION *****

      FREFRCT = SINPK1/SINPSI2

C *****

      Q22NUM = SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)

```

```

      COSPSPK2 = (COSPHI2+FMACH2)/Q22NUM
      ANGPK2 = ACOS(COSPSPK2)

C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 2, PSIPK2 (DEGREES)
      PSIPK2 = ANGPK2*180.0/PI

C GROUP VELOCITY VECTOR ANGLE CHANGE, REG. 1 TO REG. 2, DELPSI (DEG.)
      DELPSI = PSIPK2-PSIPK1
C *****

      SIN2PK2 = SIN(PSIPK2*PI/180.0)

C ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
      PSQPK = PSQCOEF*FREFRCT*ETA*COFRAT(J)/(2.0*BETA1)
C *****

C ***** START SORTING INTO REGIMES TO HANDLE ANGLES BEYOND PEAK *****
C
      COF = COFRAT(J)
      IREG = 0
      ETAC1 = 0.6*BETA1/(1.0-COFINV)
      IF(ETA.GT.ETAC1) THEN
        IREG = 1

C ***** REGION 1, SLOPE AT ARG=PI/2 USED TO PROJECT BEYOND PEAK

      EPS = 1.0/(BETA1*COF)+0.5/ETA
      EPSQ = EPS**2
      QNUM = 1.0+FMSQ1*EPSQ
      DEPDPSI = QNUM*SQRT(1.0-FM11*EPSQ)
      PSQRATC1 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
      FDEN = CFBTINSQ-EPSQ
      DPSQDPSI = (FDEN+4.0*EPSQ)*PSQRATC1*DEPDPSI/(EPS*FDEN)
      SINPSIC1 = EPS/SQRT(QNUM)
      PSIC = ASIN(SINPSIC1)*180.0/PI
      AC = ALOG(PSQRATC1)
      BC = 0.8889*DPSQDPSI/PSQRATC1
      BC = BC*PI/180.0
      CC = -0.1781*BC
      GO TO 50
      END IF

      ETAC2 = 0.6*BETA1*COF
      IF(ETA.GT.ETAC2) THEN
        IREG = 2

C ***** REGION 2, PSQratio AT ARG=-PI/2 USED TO FIT EXPONENTIAL FOR
C USE BEYOND PEAK

      EPS = 1.0/(BETA1*COF)-0.5/ETA
      PSQRATC2 = (4.0/PI)**2*EPS*COFBETIN/(COFBETIN+EPS)**2
      EPSQ = EPS**2
      QDEN = 1.0+FMSQ1*EPSQ
      SINPSIC2 = EPS/SQRT(QDEN)
      PSIC2 = ASIN(SINPSIC2)*180.0/PI
      AC = ALOG(PSQRATC2)/(PSIPK1-PSIC2)**2
      GO TO 50
      END IF

C ***** REGION 3, LOW ETA REGION, PSIPK1>60 DEG. FIT EXPONENTIAL AT

```

```

C      0.5*PSIPK1 FOR USE BEYOND PEAK.  CODE IREG=3.  EXPONENTIAL EQUATION
C      USED IN PSQ SUBROUTINE FOR PSI > PSIPK1.

      IF(PSIPK1.GT.60.0) THEN

      IREG = 3

      ANGF = 0.5*PSIPK1
      ANGRAD = ANGF*PI/180.0
      SINF = SIN(ANGRAD)
      EPS = SINF/SQRT(1.0-FMSQ1*SINF**2)
      ARG = PI*ETA*(COFBETIN-EPS)
      SINARG = SIN(ARG)
      PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
      PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
      AC = ALOG(PSQRATC3)/(PSIPK1-ANGF)**2
      GO TO 50
      END IF

C ***** REGION 4, LOW ETA REGION, PSIPK1<60 DEG. FIT EXPONENTIAL AT
C      80 deg. FOR USE BEYOND PEAK.  CODE IREG=4.  EXPONENTIAL EQUATION
C      USED IN PSQ SUBROUTINE FOR PSI>80 deg.

      IREG = 4

      ANGF = 80.0
      ANGRAD = ANGF*PI/180.0
      SINF = SIN(ANGRAD)
      EPS = SINF/SQRT(1.0-FMSQ1*SINF**2)
      ARG = PI*ETA*(COFBETIN-EPS)
      SINARG = SIN(ARG)
      PSQRATC3 = 4.0*EPS*COFBETIN/(COFBETIN+EPS)**2
      PSQRATC3 = PSQRATC3*(SINARG/ARG)**2
      AC = ALOG(PSQRATC3)/(PSIPK1-ANGF)**2

50 CONTINUE

C
C ***** FINISHED WITH 4 REGION DEFINITIONS, START PSQ CALCULATION

      DO 25 I=1,NANGLE
      FI = I
      ANGDEG2 = ANGLE(I)

      IF(ANGDEG2.EQ.0.0.AND.ISIDELN.EQ.1) THEN
      PSQRADT(I) = 0.0
      PSQTOT(I) = 0.0
      PSQTLOS(I) = 0.0
      GO TO 25
      END IF

C
      ANGDEG1 = ANGDEG2-DELPSI
      ANG = ANGDEG1
      ANGRAD1 = ANGDEG1*PI/180.0
      IF(ANGDEG1.LT.0.0) GO TO 25
      SINANG = SIN(ANGRAD1)
      COSANG = COS(ANGRAD1)

      Q1DEN = SQRT(1.0-FMSQ1*SINANG**2)
      Q1 = SINANG/Q1DEN
      ARG = PI*ETA*(Q1-COFBETIN)
      SINSQNUM = (SIN(ARG))**2

```

```

      GG = (1.0+COSANG/Q1DEN)**2/GDEN
      PSQRAT = 4.0*Q1/(BETA1*COFRAT(J)*(Q1+COFBETIN)**2)

      PSQDEN = ARG**2
      ANGCK = PSIPK1+1.0
      IF(ANG.GT.ANGCK) GO TO 6
C ***** CHECK FOR 0/0, USE LIMITING CASE IF INDETERMINATE *****
C ***** DON'T CHECK IF ANGLE ABOVE PEAK ANGLE *****
      IF(PSQDEN.LT.1.E-06) GO TO 39
      6 CONTINUE

      IF(ANG.LT.PSIPK1) GO TO 38
      IF(ANG.GE.PSIC.AND.IREG.EQ.1) THEN

C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
C ** USES SLOPE AT ARG = PI/2 TO EXTRAPOLATE TO HIGH ANGLES *****
      DANG = ANG-PSIC
      QEXP = AC+BC*DANG/(1.0+CC*DANG)
      IF(QEXP.LT.-20.) QEXP=-20.

      PSQRAT = EXP(QEXP)
      GO TO 39
      END IF

      IF(ANG.GE.PSIPK1.AND.IREG.EQ.2) THEN

C ** FOLLOWING TO REMOVE THE P**2 OSCILLATIONS BEYOND THE PEAK *****
C ** USES EXPONENTIAL FIT AT ARG = -PI/2 TO EXTRAPOLATE TO HIGH ANGLES
      QEXP = AC*(ANG-PSIPK1)**2
      IF(QEXP.LT.-20.) QEXP=-20.

      PSQRAT = EXP(QEXP)
      GO TO 39
      END IF

      IF(ANG.GE.PSIPK1.AND.IREG.EQ.3) THEN

C ** ONE OF THE LOW FREQUENCY REGIONS, REGION 3. FOLLOWING ALLOWS
C ** REASONABLE EXTRAPOLATION BEYOND 90 deg, STARTING AT PEAK.
C ** USES EXPONENTIAL FIT AT PSI = PSIPK1 AND 0.5*PSIPK1. PSIPK1>60 deg
      QEXP = AC*(ANG-PSIPK1)**2
      IF(QEXP.LT.-20.) QEXP=-20.

      PSQRAT = EXP(QEXP)
      GO TO 39
      END IF

C ** ONLY REGION LEFT, REGION 4, WITH PSIPK1<60 deg.
C ** ONE OF THE LOW FREQUENCY REGIONS. FOLLOWING ALLOWS REASONABLE
C ** EXTRAPOLATION BEYOND 90 deg, STARTING THIS TIME AT 80 deg.
C ** USES EXPONENTIAL FIT AT PSI = PSIPK1 AND 80 deg. PSIPK1<60 deg.
      IF(ANG.LT.80.0) GO TO 38

      QEXP = AC*(ANG-PSIPK1)**2
      IF(QEXP.LT.-20.) QEXP=-20.

      PSQRAT = EXP(QEXP)
      GO TO 39

      38 PSQRAT = PSQRAT*SINSQNUM/PSQDEN

```

```

39 CONTINUE

      PSQ = PSQRAT*PSQPK*GG
      PSQRAD = PSQ
C
      RAD = DISTANCE
      IF (ISIDELN.EQ.1) THEN
      RAD = DISTANCE/SIN(ANGDEG2*PI/180.0)
      END IF

      PSQ = PSQ/RAD**2

C
      PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
      PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
      PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF

C ***** NOTE THAT A TRANSMISSION LOSS (TLCF) HAS BEEN USED *****

25 CONTINUE
26 CONTINUE
   GO TO 70

45 CONTINUE
C ***** IN PLANE-WAVE CALCULATION PROCEDURE !!!!!!!!!!!!!!!!!!!!!!!

      GDEN = 4.0

      IF (FMACH1.EQ.FMACH2) THEN
      DELPSI = 0.0
      GO TO 27
      END IF

      COSPK1 = 1.0
      ANGPK1 = 0.0
C ***** GROUP VELOCITY VECTOR ANGLE IN REGION 1, PSIPK1 (DEGREES)
      PSIPK1 = 0.0
      SINPK1 = 0.0
C ***** PHASE VELOCITY VECTOR ANGLE FOR PEAK IN REGION 1 (JET)
      SIN2 = SINPK1**2
      COSPHI1 = 1.0
      PHI1RAD = 0.0
      PHI1DEG = 0.0
      SINPHI1 = 0.0
C
C ***** PHASE VELOCITY ANGLE IN REGION 2 (SURROUNDINGS)

      COSPHI2 = 1.0/(CRAT+CRAT*FMACH1-FMACH2)
      PHI2RAD = ACOS(COSPHI2)
      PHI2DEG = PHI2RAD*180.0/PI
      SINPHI2 = SIN(PHI2RAD)
      COSPSI2 = (COSPHI2+FMACH2)/SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)
      PSI2RAD = ACOS(COSPSI2)
      PSI2DEG = PSI2RAD*180.0/PI
      SINPSI2 = SIN(PSI2RAD)

      Q22DEN = SQRT(1.0+FMSQ2+2.0*FMACH2*COSPHI2)

      COSPSPK2 = (COSPHI2+FMACH2)/Q22DEN
      ANGPK2 = ACOS(COSPSPK2)
      PSIPK2 = ANGPK2*180.0/PI

```

```

C  GROUP VELOCITY VECTOR ANGLE CHANGE, REG. 1 TO REG. 2, DELPSI (DEG.)
      DELPSI = PSIPK2-PSIPK1
C  *****
C
C  27 CONTINUE

C  ***** P**2 PRINCIPAL LOBE PEAK FOR 1 WATT POWER INPUT *****
      PSQPK = 2.0*PSQCOEFP
C  *****

C  **** PSI10 BELOW IS SMALLEST ANGLE WHERE PLANE WAVE RADIATION, P**2=0
C  **** IF PSI10 > 90, PSI10 = 90 IS USED
      SINPSI10 = 1.0/SQRT(ETA**2+FMACH1**2)
      IF(SINPSI10.LT.1.0) THEN
        ANG10 = ASIN(SINPSI10)
        PSI10 = ANG10*180.0/PI
        COSPSI10 = COS(ANG10)
        GO TO 28
      END IF
      ANG10 = PI/2.0
      PSI10 = 90.0
      SINPSI10 = 1.0
      COSPSI10 = 0.0
C  28 CONTINUE

C  ***** LOW ETA REGION. FIT EXPONENTIAL AT PSICRPL FOR BEYOND PEAK.
C  ***** IF PSICRPL < 90 CAN'T BE OBTAINED, USE PSICRPL = 90.
      ANGF = 90.0
      ETACRPL = 0.5*BETA1
      SINCRPL = 1.0/SQRT(4.0*ETA**2+FMACH1**2)
      IF(ETA.GT.ETACRPL) ANGF=ASIN(SINCRPL)*180./PI
      PSICRPL = ANGF
      ANGFRAD = ANGF*PI/180.0
      SINF = SIN(ANGFRAD)
      ARG = PI*ETA*SINF/SQRT(1.0-FMSQ1*SINF**2)
      SINARG = SIN(ARG)
      PSQRATPL = (SINARG/ARG)**2
      ACPL = ALOG(PSQRATPL)/ANGF**2
C  ***** ACPL IS THE AMPLITUDE OF THE EXPONENTIAL AT PSICRPL *****
C  ***** END LOW ETA REGION EXPONENTIAL FIT AT PSICRPL. *****

C  CALCULATE P**2 AMPLITUDE CHANGE DUE TO ANGLE SHIFT FROM PSI = 0 TO
C  PSI = PSIPK2 FOR THE PLANE WAVE PASSING THROUGH THE JET SHEAR LAYER

      PSQPKMUL = 1.0
      AREA1 = 1.0-COSPSI10
      AREA2 = 1.0+SINPSI2*SINPSI10-COSPSI2*COSPSI10

      IF(PSIPK2.GT.0.0.AND.PSI10.LT.PSIPK2) THEN
        AREA2 = 2.0*SINPSI2*SINPSI10
      END IF

      PSQPKMUL = AREA1/AREA2
      PSQPK = PSQPK*PSQPKMUL

      CKPSI0 = -PSI10
      SUMPSQ = 0.0

      DO 40 I=1,NANGLE

```

```

        FI = I
        ANGDEG2 = ANGLE(I)
        IF(ANGDEG2.EQ.0.0.AND.ISIDELN.EQ.1) THEN
        PSQRADT(I) = 0.0
        PSQTOT(I) = 0.0
        PSQTLOS(I) = 0.0
        GO TO 40
        END IF
C
        ANGRAD2 = ANGDEG2*PI/180.0
        ANGDEG1 = ANGDEG2-DELPsi
        ANGRAD1 = ANGDEG1*PI/180.0

        IF(ANGDEG1.LT.CKPSI0) GO TO 40

        SINANG = SIN(ANGRAD1)
        COSANG = COS(ANGRAD1)

        Q1DEN = SQRT(1.0-FMSQ1*SINANG**2)
        Q1 = SINANG/Q1DEN
        ARG = PI*ETA*Q1
        SINSQNUM = (SIN(ARG))**2
        GG = (1.0+COSANG/Q1DEN)**2/GDEN
        PSQRAT = 1.0

        PSQDEN = ARG**2
        IF(PSQDEN.LT.1.E-06.AND.ANGDEG1.LE.90.0) GO TO 49

C ** LOW FREQUENCY REGION. FOLLOWING ALLOWS REASONABLE
C ** EXTRAPOLATION BEYOND 90 deg, STARTING AT ARG = PI/2 IF ETA>ETAcrit
C ** OR AT 90 deg. OTHERWISE. USES EXPONENTIAL FIT AT PSI = PSIPK1
C ** AND PSI = PSIFIT.

        IF(ANGDEG1.LT.PSICRPL) GO TO 48

        QEXP = ACPL*(ANGDEG1)**2
        IF(QEXP.LT.-20.) QEXP=-20.

        PSQRAT = EXP(QEXP)
        GO TO 49

48 PSQRAT = PSQRAT*SINSQNUM/PSQDEN
49 CONTINUE

        PSQ = PSQRAT*PSQPK*GG

        PSQRAD = PSQ
C
        RAD = DISTANCE
        IF(ISIDELN.EQ.1) THEN
        RAD = DISTANCE/SIN(ANGDEG2*PI/180.0)
        END IF

        PSQ = PSQ/RAD**2
C
        PSQRADT(I) = PSQRADT(I)+POWCOEF*PSQRAD
        PSQTOT(I) = PSQTOT(I)+POWCOEF*PSQ
        PSQTLOS(I) = PSQTLOS(I)+POWCOEF*PSQ*TLCF
C ***** NOTE THAT A TRANSMISSION LOSS (TLCF) HAS BEEN USED *****

```

```

40 CONTINUE
41 CONTINUE
C
70 CONTINUE

      FNANGLE = NANGLE
      SUMWATT = 0.0
      DO 75 I=1,NANGLE
      ANGRAD = ANGLE(I)*PI/180.0
      SUMWATT = SUMWATT+PSQRADT(I)*SIN(ANGRAD)

C ** CONVERTING TO ANGLE FROM INLET AXIS FOR THIS AFT RADIATED NOISE **

      ANGLE(I) = 180.0-ANGLE(I)
      IF(PSQTOT(I).LT.4.E-08) THEN
      SPLTL(I) = 20.0
      SPL(I) = 20.0
      GO TO 75
      END IF
      SPLTL(I) = 10.0*ALOG10(PSQTLOS(I))+93.9794
      SPL(I) = 10.0*ALOG10(PSQTOT(I))+93.9794
75 CONTINUE

      WATTINT = SUMWATT*PI/(POWCON*FNANGLE)
      SPLDIF = 10.0*ALOG10(WATTS/WATTINT)

      DO 80 I=1,NANGLE
      SPLTL(I) = SPLTL(I)+SPLDIF
      SPL(I) = SPL(I)+SPLDIF
80 CONTINUE

      RETURN
      END

C
C *****
C ***** END OF MAIN SUBROUTINE "BBRDCFCR" *****
C ***** ALTERED 02/19/1998, E. J. RICE *****
C *****
C
C
C *****
C SUBROUTINE FOR CALC NOZZLE CONDITIONS, FINAL "CORE" JET MACH NUMBER,
C VELOCITY, AND DIAMETER. ** NOTE ** ASSUME ANNULAR CORE DUCT AND
C CIRCULAR CORE NOZZLE.
C
      SUBROUTINE CORNOZ(TTCOR,PTCOR,PSTS,HTRATC,ANOZRATC,DDUCTC,FMACHC,
1FMACH1,CJET,DJET,TJET,PNOZ,DNOZ,CNOZ,FMACHN)

C
      PI = 3.1415927
      QM = 1.0+0.2*FMACHC**2
      DIN = DDUCTC*HTRATC
C ***** ANNULAR CORE DUCT *****
      ADUCT = PI*(DDUCTC**2-DIN**2)/4.0
      ANOZ = ADUCT*ANOZRATC
C ***** CIRCULAR CORE NOZZLE *****
      DNOZ = SQRT(4.0*ANOZ/PI)
      TDUCT = TTCOR/QM
      VSOND = 49.0421*SQRT(TDUCT)

```

```

      VDUCT = FMACHC*VSOND
      RHOT = 144.0*PTCOR/(53.3*TTCOR)
      RHOD = RHOT/QM**2.5
      FMASS = RHOD*ADUCT*VDUCT
C ***** NOTE - AREAS ARE IN SQUARE INCHES *****
C
C ***** SOLVE FOR NOZZLE MACH NUMBER *****
C
      QQDUCT = FMACHC/QM**3/ANOZRATC
      FN = 1.0
      DIFP = FN/(1.0+0.2*FN**2)**3-QQDUCT
      FNP = 1.0

      DO 10 I=1,50
      FN = 0.975*FN
      DIF = FN/(1.0+0.2*FN**2)**3-QQDUCT
C      WRITE(3,100) I,FN,DIF
C 100 FORMAT(/,' I=',I2,' FN=',F7.4,' DIF=',1PE9.2)
      IF(DIF.LE.0.0) GO TO 12
      FNP = FN
      DIFP = DIF
10 CONTINUE
      FN = QQDUCT
111 DO 11 I=1,10
      FN = QQDUCT*(1.0+0.2*FN**2)**3
C      WRITE(3,102) I,FN
C 102 FORMAT(/,' REFINED, I=',I2,' FN=',F7.4)

11 CONTINUE
      GO TO 14
12 CONTINUE
      FN = (DIFP*FN-DIF*FNP)/(DIFP-DIF)
C      WRITE(3,101) FN
C 101 FORMAT(/,' INTERPOLATED FN =',F7.4)

      GO TO 111
14 CONTINUE
      FMACHN = FN
C      WRITE(3,103) FMACHN
C 103 FORMAT(/,' FINAL ***** FMACHN =',F7.4)

C ***** "FMACH1" FINAL JET VELOCITY *****

      FMACH1 = SQRT(5.0*((PTCOR/PSTS)**(2./7.)-1.0))
      QNTJ = 1.0+0.2*FMACH1**2
      TJET = TTCOR/QNTJ
      RHOJ = RHOT/QNTJ**2.5
      CJET = 49.0421*SQRT(TJET)
      VJET = CJET*FMACH1
      AJET = FMASS/(RHOJ*VJET)
      DJET = SQRT(4.0*AJET/PI)

C ** FOLLOWING NOT USED HERE BUT MIGHT BE HANDY FOR LATER USE *****
C ** CHECK NOZZLE THROAT PRESSURE VRS. SURROUNDING AMBIENT PRESSURE **
      QMN = 1.0/(1.0+0.2*FMACHN**2)
      PNOZ = PTCOR*QMN**3.5
      TNOZ = TTCOR*QMN
      CNOZ = 49.0421*SQRT(TNOZ)
      RETURN
      END
C

```

```

C *****
C ***** END OF SUBROUTINE "CORNOZ" *****
C *****
C
C *****
C ***** SUBROUTINE 'MATPRD33' *****
C *****
C
C YIELDS PRODUCT OF TWO COMPLEX 3 BY 3 MATRICES (A,B) AS A COMPLEX
C 3 BY 3 COMPLEX MATRIX "C."
C
C SUBROUTINE MATPRD33 ( A,B,C )
C COMPLEX A(3,3),B(3,3),C(3,3)
C
C DO 300 I = 1,3
C   DO 200 J = 1,3
C     C(I,J) = CMPLX(0.00,0.00)
C     DO 100 K = 1,3
C       C(I,J) = C(I,J)+A(I,K)*B(K,J)
100     CONTINUE
200   CONTINUE
300 CONTINUE
C
C RETURN
C END
C
C *****
C ***** END OF SUBROUTINE 'MATPRD33' *****
C *****
C
C *****
C ***** SUBROUTINE 'VECPRDS' *****
C *****
C
C YIELDS PRODUCT OF COMPLEX VECTOR BY A COMPLEX SCALAR.
C
C SUBROUTINE VECPRDS ( A,B,C )
C COMPLEX A(3),B,C(3)
C
C DO 100 I = 1,3
C   C(I) = A(I)*B
100 CONTINUE
C
C RETURN
C END
C
C *****
C ***** END OF SUBROUTINE 'VECPRDS' *****
C *****
C
C *****
C ***** SUBROUTINE 'MPRD34' *****
C *****
C
C INPUT MATRICES "A,B" ARE COMPLEX "3 BY 4" MATRICES."A" IS PART
C OF A "4 BY 4" MATRIX (SAY "AO") WHOSE 4TH ROW HAS FIRST THREE COLUMNS AS
C ZERO AND FOURTH COLUMN AS "1." "B" IS SIMILAR TO "A" EXCEPT THAT FOURTH
C ROW, FOURTH COLUMN IS "Z." "C" IS THE COMPLEX (3 BY 4) PART OF THE
C PRODUCT "AO*BO." 4TH ROW, 4TH COLUMN OF "CO" WOULD BE "Z" AND FIRST 3
C COLUMNS OF 4TH ROW OF "CO" WOULD BE ZERO.

```

```

C
      SUBROUTINE MPRD34 ( A,B,Z,C )
      COMPLEX A(3,4),B(3,4),Z,C(3,4),TTA(3,3),T1A(3),TTB(3,3),
&          T1B(3),TTC(3,3),T1CP(3),T1C(3)
C
      DO 200 I = 1,3
        DO 100 J = 1,3
          TTA(I,J) = A(I,J)
          TTB(I,J) = B(I,J)
100      CONTINUE
200      CONTINUE
C
      DO 300 I = 1,3
        T1A(I) = A(I,4)
        T1B(I) = B(I,4)
300      CONTINUE
C
      CALL MATPRD33 ( TTA,TTB,TTC )
      CALL MATPRD ( TTA,T1B,T1CP )
      DO 400 I = 1,3
        T1C(I) = T1CP(I)+Z*T1A(I)
400      CONTINUE
C
      DO 600 I = 1,3
        DO 500 J = 1,3
          C(I,J) = TTC(I,J)
500      CONTINUE
600      CONTINUE
C
      DO 700 I = 1,3
        C(I,4) = T1C(I)
700      CONTINUE
C
      RETURN
      END
C
C          *****
C ***** END OF SUBROUTINE 'MPRD34' *****
C          *****
C
C          *****
C ***** SUBROUTINE 'SHIFT' *****
C          *****
C
C      GIVEN AN INPUT COMPLEX (3 BY 4) MATRIX "A"
C      AND THREE INPUT COMPLEX CONSTANTS "R1,R2,R3" , RETURNS
C      A COMPLEX (3 BY 4) MATRIX "B" WHOSE FIRST,SECOND AND THIRD ROWS ARE THE
C      CORRESPONDING ROWS OF "A" MULTIPLIED BY "R1,R2,R3."
C
      SUBROUTINE SHIFT ( A,R1,R2,R3,B )
      COMPLEX A(3,4),B(3,4),R1,R2,R3,R(3)
C
      R(1)      = R1
      R(2)      = R2
      R(3)      = R3
C
      DO 200 I = 1,3
        DO 100 J = 1,4
          B(I,J) = A(I,J)*R(I)
100      CONTINUE

```

```

200 CONTINUE
C
      RETURN
      END
C
C
C *****
C ***** END OF SUBROUTINE 'SHIFT' *****
C *****
C
C *****
C ***** SUBROUTINE 'CHOKE' *****
C *****
C
C      GIVEN THE NUMBER OF CHOKED BLADE ROWS "NCHR" AND THE SPECIFIC
C      VECTOR "ICH" OF THE NUMBERS OF THE CHOKED BLADE ROWS, DETERMINES
C      IF INPUT ROW NUMBER "IR" IS ONE OF THE CHOKED ROWS. "ICHOKE" IS
C      SET AS ZERO OR UNITY DEPENDING ON IF ROW "IR" IS UNCHOKED OR CHOKED.
C
      SUBROUTINE CHOKE ( NCHR,IR,ICH,ICHOKE )
      DIMENSION ICH(20)
C
      ICHOKE = 0
      DO 100 I = 1,NCHR
        IF ( IR.EQ.ICH(I) ) ICHOKE = 1
100 CONTINUE
C
      RETURN
      END
C
C *****
C ***** END OF SUBROUTINE 'CHOKE' *****
C *****
C *****
C ***** SUBROUTINE 'VRM33P' *****
C *****
C
C      YIELDS PRODUCT OF ROW VECTOR 'A' AND '3 BY 3' MATRIX 'B' AND
C      RESULT IS THE ROW VECTOR 'C'
C
      SUBROUTINE VRM33P ( A,B,C )
      COMPLEX A(3),B(3,3),C(3)
C
      DO 200 I = 1,3
        C(I) = CMPLX(0.00,0.00)
        DO 100 J = 1,3
          C(I) = A(J)*B(J,I)+C(I)
100 CONTINUE
200 CONTINUE
C
      RETURN
      END
C
C *****
C ***** END OF SUBROUTINE 'VRM33P' *****
C *****
C *****
C ***** SUBROUTINE 'SMATCH' *****
C *****
C

```

```

C      THIS SUBROUTINE SETS UP THE ROW VECTOR 'BUCH' WHOSE FOUR ELEMENTS
C      ARE 'INFLUENCE COEFFICIENTS' REPRESENTING THE 'CHOKING CONDITION'
C      FOR THE BLADE ROW AT THE DOWNSTREAM END FOR WAVE SYSTEMS AS TABULATED
C      BELOW:
C
C      DNSTR SOUND(1)      UPSTR SOUND(2)      SHEAR(3)      ENTROPY(4)
C
C      SUBROUTINE SMATCH ( CR,G,KIN,KY,MTDNS,MX,MY,BUCH )
C
C      COMPLEX BUCH(4),KX(2),KXI,APCH,U,V,P
C      REAL KIN,KY,MTDNS,MX,MY,K,KXSH,MTD,MRELD
C
C      K          = KIN/CR
C      MTD         = MTDNS/CR
C      MRELD       = SQRT(MX**2+(MY-MTD)**2)
C      TMREL       = 1.-MRELD**2
C
C      NOTATION FOR "BUCH(I)" WITH "I=1,4"
C      I = 1 , DOWNSTREAM SOUND
C      I = 2 , UPSTREAM SOUND
C      I = 3 , SHEAR
C      I = 4 , ENTROPY
C
C      CALL WNUMB ( K,KY,MX,MY,KX,KXSH )
C      CALCULATE "BUCH"
C      KXI         = KX(1)
C      CALL UVP ( 1,K,KXI,KXSH,KY,MX,MY,P,U,V )
C      CALL CHOKP ( G,MTD,MX,MY,U,V,P,APCH )
C      BUCH(1)     = APCH
C
C      KXI         = KX(2)
C      CALL UVP ( 2,K,KXI,KXSH,KY,MX,MY,P,U,V )
C      CALL CHOKP ( G,MTD,MX,MY,U,V,P,APCH )
C      BUCH(2)     = APCH
C
C      KXI         = CMPLX(0.00,0.00)
C      CALL UVP ( 3,K,KXI,KXSH,KY,MX,MY,P,U,V )
C      CALL CHOKP ( G,MTD,MX,MY,U,V,P,APCH )
C      BUCH(3)     = APCH
C
C      BUCH(4)     = CMPLX(0.50,0.00)*TMREL
C
C      RETURN
C      END
C
C      *****
C      ***** END OF SUBROUTINE 'SMATCH' *****
C      *****
C
C      *****
C      ***** SUBROUTINE 'INV2' *****
C      *****
C
C      SOLVES "A*X=Y" WHERE "A" IS A COMPLEX "2 BY 2" MATRIX AND "X,Y"
C      ARE 2 ELEMENT COMPLEX COLUMN VECTORS."A,Y" ARE INPUTS AND "X" IS
C      THE OUTPUT.
C
C      SUBROUTINE INV2 ( A,Y,X )
C      COMPLEX A(2,2),Y(2),X(2),DET
C

```

```

      DET      = DETM(A)
C
      X(1)     = (A(2,2)*Y(1)-A(1,2)*Y(2))/DET
      X(2)     = (A(1,1)*Y(2)-A(2,1)*Y(1))/DET
C
      RETURN
      END
C
C      *****
C***** END OF SUBROUTINE 'INV2' *****
C      *****
C
C      *****
C***** SUBROUTINE 'INV3' *****
C      *****
C
C      SOLVES "A*X=Y" WHERE "A" IS A COMPLEX "3 BY 3" MATRIX AND "X,Y"
C      ARE 3 ELEMENT COMPLEX COLUMN VECTORS."A,Y" ARE INPUTS AND "X" IS
C      THE OUTPUT.
C
      SUBROUTINE INV3 ( A,Y,X )
      COMPLEX A(3,3),Y(3),X(3),AI(3,3)
C
      CALL MATINV ( A,AI )
      CALL MATPRD ( AI,Y,X )
C
      RETURN
      END
C
C      *****
C***** END OF SUBROUTINE 'INV3' *****
C      *****
C
C      *****
C***** SUBROUTINE 'MSFL' *****
C      *****
C
C      CALCULATES MASS FLUX AND STAGNATION ENTHALPY FLUX GIVEN COMPLEX
C      AMPLITUDES OF DNSTR SOUND,UPSTR SOUND,SHEAR AND ENTROPY WAVES (AS
C      A(1),A(2),A(3) AND 'ENT') AND (2 BY 4) INFLUENCE COEFFICIENT MATRIX
C      'B'
C
      SUBROUTINE MSFL ( A,ENT,B,MF,SF )
      COMPLEX A(3),ENT,B(2,4),MF,SF
C
      MF = CMPLX(0.0,0.0)
      SF = CMPLX(0.0,0.0)
C
      DO I = 1,3
         MF = MF+A(I)*B(1,I)
         SF = SF+A(I)*B(2,I)
      END DO
      MF = MF+ENT*B(1,4)
      SF = SF+ENT*B(2,4)
      RETURN
      END
C
C      *****
C***** END OF SUBROUTINE 'MSFL' *****
C      *****

```

```

C
C
C *****
C ***** SUBROUTINE 'MPR4' *****
C *****
C
C CALCULATES FIRST THE TRIPLE PRODUCT "BDI*BU*SU" WHERE 'BDI,BU,SU'
C HAVE THEIR USUAL SIGNIFICANCE AND THEN SHIFTS ROWS BY 'R1,R2,R3'.
C
C SUBROUTINE MPR4 ( BDI,BU,SU,S4,R1,R2,R3,RES )
C COMPLEX BDI(3,4),BU(2,4),SU(3,4),S4,R1,R2,R3,RES(3,4),
C & TRES(3,4),TRES1(3,4)
C
C CALL MATPRD1 ( BDI,BU,TRES )
C CALL MPRD34 ( TRES,SU,S4,TRES1 )
C CALL SHIFT ( TRES1,R1,R2,R3,RES )
C
C RETURN
C END
C
C *****
C ***** END OF SUBROUTINE 'MPR4' *****
C *****
C
C *****
C ***** SUBROUTINE 'EQAT' *****
C *****
C
C EQUATES TWO COMPLEX (3 BY 4) MATRICES.
C
C SUBROUTINE EQAT ( A,B )
C COMPLEX A(3,4),B(3,4)
C
C DO I = 1,3
C   DO J = 1,4
C     B(I,J) = A(I,J)
C   END DO
C END DO
C
C RETURN
C END
C
C *****
C ***** END OF SUBROUTINE 'EQAT' *****
C *****
C
C *****
C ***** SUBROUTINE 'EQAT1' *****
C *****
C
C EQUATES TWO COMPLEX (2 BY 4) MATRICES.
C
C SUBROUTINE EQAT1 ( A,B )
C COMPLEX A(2,4),B(2,4)
C
C DO I = 1,2
C   DO J = 1,4
C     B(I,J) = A(I,J)
C   END DO
C END DO
C

```

```

      RETURN
      END

C
C
C *****
C ***** END OF SUBROUTINE 'EQAT1' *****
C *****
C
C *****
C ***** SUBROUTINE 'DIAG' *****
C *****
C
C      SETS DIAGONAL ELEMENTS OF A COMPLEX ( 3 BY 4 ) MATRIX TO R1,R2,R3
C      RESPECTIVELY AND ALL OFF DIAGONAL TERMS TO ZERO.'D4' IS SET TO R3
C
C      SUBROUTINE DIAG ( R1,R2,R3,D,D4 )
C      COMPLEX D(3,4),R1,R2,R3,R(3),D4
C
C      R(1) = R1
C      R(2) = R2
C      R(3) = R3
C      D4 = R3
C      DO I = 1,3
C         DO J = 1,4
C            IF ( I.NE.J ) D(I,J) = CMPLX ( 0.00,0.00 )
C            IF ( I.EQ.J ) D(I,J) = R(I)
C         END DO
C      END DO
C
C      RETURN
C      END

C
C *****
C ***** END OF SUBROUTINE 'DIAG' *****
C *****
C
C *****
C ***** SUBROUTINE 'CHSOL' *****
C *****
C
C      SOLVES FOR WAVE SYSTEMS WHEN A CHOKED ROW IS ENCOUNTERED BEYOND
C      THE FIRST TIME AND COMPUTES NEW MASS FLUX AND STAGNATION ENTHALPY FLUX.
C
C      SUBROUTINE CHSOL ( BC,SU,S4,SD,BU,ENT,EWAVE,MFL,SEFL,
C      & MFLN,SEFLN )
C      COMPLEX BC(4),SU(3,4),ENT,MFL,MFLN,SEFL,SEFLN,P(4),SD(3,4),
C      & RHS(3),SOL(3),SOLN(3),NEW(3,3),S4,EWAVE,BU(2,4)
C
C      DO J = 1,4
C         P(J) = CMPLX(0.00,0.00)
C         DO I = 1,3
C            P(J) = P(J)+BC(I)*SU(I,J)
C         END DO
C      END DO
C      P(4) = P(4)+BC(4)*S4
C
C      DO I = 1,3
C         IF ( I.LT.3 ) THEN
C            DO J = 1,3
C               NEW(I,J) = SD(I,J)
C            END DO

```

```

        ELSE
            DO J = 1,3
                NEW(I,J) = P(J)
            END DO
        END IF
    END DO
C
    RHS(1) = MFL-SD(1,4)*ENT
    RHS(2) = SEFL-SD(2,4)*ENT
    RHS(3) = -P(4)*ENT
C
    CALL INV3 ( NEW,RHS,SOL )
C
    DO I = 1,3
        SOLN(I) = CMPLX(0.00,0.00)
        DO J = 1,3
            SOLN(I) = SOLN(I)+SU(I,J)*SOL(J)
        END DO
        SOLN(I) = SOLN(I)+SU(I,4)*ENT
    END DO
C
    CALL MSFL ( SOLN,EWAVE,BU,MFLN,SEFLN )
C
    RETURN
    END
C
C *****
C ***** END OF SUBROUTINE 'CHSOL' *****
C *****
C
C *****
C ***** SUBROUTINE 'DISOL' *****
C *****
C
C SOLVES FOR WAVE SYSTEMS AT DISCHARGE WHEN LAST BLADE ROW IS NOT
C CHOKED.
C
    SUBROUTINE DISOL ( SU,SD,ENT,MFL,SEFL,ADSW )
    COMPLEX SU(3,4),ENT,P(4),SD(3,4),MFL,SEFL,
&      RHS(3),SOL(3),SOLN(3),NEW(3,3)
C
    DO J = 1,4
        P(J) = SU(2,J)
    END DO
C
    DO I = 1,3
        IF ( I.LT.3 ) THEN
            DO J = 1,3
                NEW(I,J) = SD(I,J)
            END DO
        ELSE
            DO J = 1,3
                NEW(I,J) = P(J)
            END DO
        END IF
    END DO
C
    RHS(1) = MFL-SD(1,4)*ENT
    RHS(2) = SEFL-SD(2,4)*ENT
    RHS(3) = -P(4)*ENT

```

```

C
      CALL INV3 ( NEW,RHS,SOL )
C
      DO I = 1,3
        SOLN(I) = CMPLX(0.00,0.00)
        DO J = 1,3
          SOLN(I) = SOLN(I)+SU(I,J)*SOL(J)
        END DO
        SOLN(I) = SOLN(I)+SU(I,4)*ENT
      END DO
C
      ADSW = CABS ( SOLN(1) )
C
      RETURN
      END
C
C *****
C ***** END OF SUBROUTINE 'DISOL' *****
C *****
C
C *****
C ***** SUBROUTINE 'DICH' *****
C *****
C
C SOLVES FOR WAVE SYSTEMS AT DISCHARGE WHEN LAST BLADE ROW IS
C CHOKED.
C
      SUBROUTINE DICH ( BD,MFL,SEFL,ENT,ADSW )
      COMPLEX BD(3,4),ENT,MFL,SEFL,LSP(2,2),LS(2),RHS(2)
C
      LSP(1,1) = BD(1,1)
      LSP(2,1) = BD(2,1)
      LSP(1,2) = BD(1,3)
      LSP(2,2) = BD(2,3)
C
      RHS(1)   = MFL-ENT*BD(1,4)
      RHS(2)   = SEFL-ENT*BD(2,4)
C
      CALL INV2 ( LSP,RHS,LS )
C
      ADSW      = CABS ( LS(1) )
C
      RETURN
C
      END
C
C *****
C ***** END OF SUBROUTINE 'DICH' *****
C *****
C
C *****
C ***** SUBROUTINE 'ANGSRT' *****
C *****
C
C GIVEN AN ANGLE SET 'ANGLE(I),I=1,NANG' AND AN ANGLE 'ANG',DETERMINES
C THE ANGLE OF SET 'ANGLE' WHICH IS CLOSEST TO 'ANG'.RETURNS THIS VALUE
C OF SET AS 'ANGO' AND ASSOCIATED 'I' AS 'IANG'.
C
      SUBROUTINE ANGSRT ( NANG,ANGLE,ANG,IANG,ANGO )
C

```

```

C      DIMENSION ANGLE ( 200 )
C
C      AMODMIN = ABS ( ANGLE(1)-ANG )
C      IANG    = 1
C      ANGO    = ANGLE(1)
C
C      DO I = 2,NANG
C        ANGMOD = ABS ( ANGLE(I)-ANG )
C        IF ( ANGMOD.LT.AMODMIN ) THEN
C          ANGO    = ANGLE(I)
C          IANG    = I
C          AMODMIN = ANGMOD
C        END IF
C      END DO
C
C      RETURN
C
C      END
C
C      *****
C***** END OF SUBROUTINE 'ANGSRT' *****
C      *****
C
C      *****
C***** SUBROUTINE 'INTSTR' *****
C      *****
C
C      INTERPOLATES "CE,PE,RMX(NS),RMY(NS),DIN(NR),DOUT(NR),CRD(NR),
C      AXV(NR-1),hOVH" FROM PITCH LINE VALUES, NON DIMENSIONAL RADIAL HEIGHT
C      AND UPSTREAM & DOWNSTREAM HUB-TIP RATIOS FOR STRIP OF INTEREST.
C
C      SUBROUTINE INTSTR ( NSTR )
C
C      REAL      MXP,MYP,M2P,MXT,MYTP,MYT,M2T,MX,MY
C
C      COMMON    CE,PE,RMX(50),RMY(50),DIN(50),DOUT(50),CRD(50),GAM,
C      &          AXV(50),RHT(25),SIGIN(50),SIGOUT(50),NR,NS,NRHT,
C      &          SMX(50),SMY(50),SDIN(50),SDOUT(50),SCRD(50),
C      &          SAXV(50),ShOVH,SANNHT,SCE,SPE
C
C      GC1      = (GAM-1.)/2.
C      GC2      = 2.*GAM/(GAM-1.)
C      SAXV(1)  = 0.00
C      SAXV(NS) = 0.00
C      SCRD(NS) = 1.00
C      PI       = 3.141593
C      SIGD     = SIGOUT(NR)
C      DD       = DOUT(NR)
C      NP1      = NSTR+1
C      NM1      = NSTR-1
C      X        = RHT(NSTR)
C
C      IF ( NSTR.EQ.1 )    DELX = 0.5*(RHT(2)+RHT(1))
C      IF ( NSTR.EQ.NRHT ) DELX = 1.-0.5*(RHT(NSTR)+RHT(NM1))
C      IF ( (NSTR.GT.1).AND.(NSTR.LT.NRHT) )
C      &      DELX = 0.5*(RHT(NP1)-RHT(NM1))
C      ShOVH    = DELX
C      SANNHT    = DD*(1.-SIGD)*DELX/(1.+SIGD)
C
C      DO CALCULATIONS FOR LAST SPACE FIRST

```

```

      RP      = (1.+SIGD)/2.
      MXP     = RMX(NS)
      MYP     = RMY(NS)
      M2P     = MXP**2+MYP**2
      RTP2    = 1./(1.+GC1*M2P)
      RTP     = SQRT(RTP2)
      MXT     = MXP*RTP
      MYTP    = MYP*RTP
      R       = SIGD+X*(1.-SIGD)
      SDOUT(NR) = 2.*DD*R/(1.+SIGD)
      RU      = SIGIN(NR)+X*(1.-SIGIN(NR))
      SDIN(NR) = 2.*DIN(NR)*RU/(1.+SIGIN(NR))
      MYT     = MYTP*RP/R
      M2T     = MYT**2+MXT**2
      RT2     = 1.-GC1*M2T
      RT      = SQRT(RT2)
      MX      = MXT/RT
      MY      = MYT/RT
      SMX(NS) = MX
      SMY(NS) = MY
      CEOCPE  = RT/RTP
      SCE     = CE*CEOCPE
      SPE     = PE*CEOCPE**GC2
C      DO CALCULATIONS FOR SPACES 1 TO (NS-1)
      NSM1    = NS-1
      DO 100 I = 2,NSM1
C      DO CALCULATIONS FOR (I-1) SPACE
      IM1     = I-1
      RP      = (1.+SIGIN(IM1))/2.
      MXP     = RMX(IM1)
      MYP     = RMY(IM1)
      M2P     = MXP**2+MYP**2
      RTP2    = 1./(1.+GC1*M2P)
      RTP     = SQRT(RTP2)
      MXT     = MXP*RTP
      MYTP    = MYP*RTP
      R       = SIGIN(IM1)+X*(1.-SIGIN(IM1))
      MYT     = MYTP*RP/R
      M2T     = MYT**2+MXT**2
      RT2     = 1.-GC1*M2T
      RT      = SQRT(RT2)
      MX      = MXT/RT
      MY      = MYT/RT
      SMX(IM1) = MX
      SMY(IM1) = MY
      SDIN(IM1) = 2.*DIN(IM1)*R/(1.+SIGIN(IM1))
      SCRD(IM1) = (RT/RTP)*CRD(IM1)/CEOCPE
      SAXV(IM1) = AXV(IM1)*RP/R
C      DO CALCULATIONS FOR SPACE "I"
      RP      = (1.+SIGOUT(IM1))/2.
      MXP     = RMX(I)
      MYP     = RMY(I)
      M2P     = MXP**2+MYP**2
      RTP2    = 1./(1.+GC1*M2P)
      RTP     = SQRT(RTP2)
      MXT     = MXP*RTP
      MYTP    = MYP*RTP
      R       = SIGOUT(IM1)+X*(1.-SIGOUT(IM1))
      MYT     = MYTP*RP/R
      M2T     = MYT**2+MXT**2

```

```

        RT2      = 1.-GC1*M2T
        RT       = SQRT(RT2)
        MX       = MXT/RT
        MY       = MYT/RT
        SMX(I)   = MX
        SMY(I)   = MY
        SDOUT(IM1) = 2.*DOUT(IM1)*R/(1.+SIGOUT(IM1))
        SCRD(I)  = (RT/RTP)*CRD(I)/CEOCPE
        SAXV(I)  = AXV(I)*RP/R
100 CONTINUE
C
        RETURN
C
        END
C
C          *****
C ***** END OF SUBROUTINE 'INTSTR' *****
C          *****
C

```

4.6 Sample Input File for Core Noise Program

```

11 12 8 15                      NR,NF,NRHT,NSPL
200.                            TOBF(1)
252.                            TOBF(2)
315.                            TOBF(3)
405.                            TOBF(4)
500.                            TOBF(5)
630.                            TOBF(6)
800.                            TOBF(7)
1000.                           TOBF(8)
1250.                           TOBF(9)
1575.                           TOBF(10)
2000.                           TOBF(11)
2520.                           TOBF(12)-TOBF(NF)
1695 1.353 1.4 15.33           CE,GAM,GAMA,PE
.2 8.50 .13 .30                RHT(1),TFPV(1),LOVRV(1),LOVR2V(1)
.3 8.0 .13 .30                 RHT(2),TFPV(2),LOVRV(2),LOVR2V(2)
.4 8.0 .13 .30                 RHT(3),TFPV(3),LOVRV(3),LOVR2V(3)
.5 8.5 .13 .30                 RHT(4),TFPV(4),LOVRV(4),LOVR2V(4)
.6 10.0 .13 .30                RHT(5),TFPV(5),LOVRV(5),LOVR2V(5)
.7 10.5 .13 .30                RHT(6),TFPV(6),LOVRV(6),LOVR2V(6)
.8 12.0 .13 .30                RHT(7),TFPV(7),LOVRV(7),LOVR2V(7)
.9 12.0 .13 .30                RHT(8),TFPV(8),LOVRV(8),LOVR2V(8)
1238 15.49 520 14.7 .70 .95 400. TTOT,PTOT,TA,PA,HTR,ANOZRAT,DISTANCE
1 38 .1 1 5.                   ISIDELN,DDUCT,FMACH2,NCOF,DELANG
90.0                           ASPL(1)
95.0                           ASPL(2)
100.0                          ASPL(3)
105.0                          ASPL(4)
110.0                          ASPL(5)
115.0                          ASPL(6)
120.0                          ASPL(7)
125.0                          ASPL(8)
130.0                          ASPL(9)
135.0                          ASPL(10)
140.0                          ASPL(11)
145.0                          ASPL(12)
150.0                          ASPL(13)
155.0                          ASPL(14)

```

```

160.0          ASPL(15)
0.0773 0.0000  RMX(1),RMY(1)
0.2196 0.9961  RMX(2),RMY(2)
0.3809 0.1438  RMX(3),RMY(3)
0.2894 0.5625  RMX(4),RMY(4)
0.2527 0.2905  RMX(5),RMY(5)
0.2046 0.4094  RMX(6),RMY(6)
0.1907 0.2126  RMX(7),RMY(7)
0.1639 0.2770  RMX(8),RMY(8)
0.1553 0.0944  RMX(9),RMY(9)
0.1441 0.1792  RMX(10),RMY(10)
0.1451 -.0321  RMX(11),RMY(11)
0.1250 -.0013  RMX(12),RMY(12)-RMX(NS),RMY(NS)
0.      27.125 27.365 1.22 .8378 .8743  S1 HPT,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
11660. 27.349 27.232 1.12 .8698 .8618  R1 HPT,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
0.      27.381 28.006 1.04 .8409 .798   S1,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
2027.  28.144 28.526 1.02 .7923 .7751  R1,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
0.      28.888 29.690 1.02 .7606 .7366  S2,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
2027.  29.953 30.435 1.02 .7311 .7203  R2,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
0.      30.864 31.463 1.01 .7101 .6945  S3,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
2027.  31.662 32.052 1.01 .6896 .6809  R3,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
0.      32.282 32.594 1.00 .675   .6657  S4,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
2027.  32.635 32.638 1.00 .6647 .6646  R4,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
0.      32.411 31.999 1.00 .6724 .687   OGV,RPM,DIN,DOUT,CRD,SIGIN,SIGOUT
0.0393        AXV(2)
0.0402        AXV(3)
0.0162        AXV(4)
0.0369        AXV(5)
0.0284        AXV(6)
0.0458        AXV(7)
0.0247        AXV(8)
0.0386        AXV(9)
0.0265        AXV(10)
0.0265        AXV(11)-AXV(NR)
25           NCOF- # of cut off ratio bins
3           ISPEC:IF EQ. 1 OR 2 EXPONENT IN CORR
FCN  $e(-(r/l)**ISPEC);3:1/(1+(r*pi/2l)**2)$ 

```

4.7 Sample Output File for Core Noise Program

***** INPUT PARAMETERS *****

```

NUMBER OF BLADE ROWS          =      11

EXIT STATIC SPD. OF SOUND ( fps ) = 1695.0
SPECIFIC HEAT RATIO OF GAS      =      1.35
SPECIFIC HEAT RATIO OF GAS-AMBIENT =      1.40
EXIT STATIC PRESSURE (psia)     =      15.33

TOTAL TEMP. IN AFT DUCT,DEG R   = 1238.0
TOTAL PRESSURE IN AFT DUCT ,psia = 15.49
AMBIENT STATIC TEMPERATURE,DEG R = 520.0
AMBIENT STATIC PRESSURE,psia    = 14.70
AFT DUCT HUB TIP RATIO          = 0.700
NOZZLE THROAT/AFT DUCT AREA RATIO = 0.950
AFT DUCT DIAMETER,inches        = 38.00
AMBIENT MACH NUMBER             = 0.100

```

TEMPERATURE FLUCTUATION RELATED INPUTS

RHT	Trms/T %	Ax L/R	Ta L/R
0.20	8.50	0.1300	0.3000
0.30	8.00	0.1300	0.3000
0.40	8.00	0.1300	0.3000
0.50	8.50	0.1300	0.3000
0.60	10.00	0.1300	0.3000
0.70	10.50	0.1300	0.3000
0.80	12.00	0.1300	0.3000
0.90	12.00	0.1300	0.3000

UPSTREAM SPACE-@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.0773
TANGENTIAL MACH NUMBER IN SPACE = 0.0000
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 0.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.2200
DIAMETER AHEAD OF BLADE ROW (ins) = 27.13
DIAMETER AFT OF BLADE ROW (ins) = 27.36
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.8378
HUB TO TIP RATIO AFT OF BLADE ROW = 0.8743

SPACE NUMBER 2 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.2196
TANGENTIAL MACH NUMBER IN SPACE = 0.9961
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 11660.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.1200
DIAMETER AHEAD OF BLADE ROW (ins) = 27.35
DIAMETER AFT OF BLADE ROW (ins) = 27.23
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.8698
HUB TO TIP RATIO AFT OF BLADE ROW = 0.8618

AXIAL SPACING OF SPACE NORMALISED BY
MEAN RADIUS OF ANNULUS = 0.0393

SPACE NUMBER 3 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.3809
TANGENTIAL MACH NUMBER IN SPACE = 0.1438
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 0.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.0400
DIAMETER AHEAD OF BLADE ROW (ins) = 27.38
DIAMETER AFT OF BLADE ROW (ins) = 28.01
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.8409
HUB TO TIP RATIO AFT OF BLADE ROW = 0.7980

AXIAL SPACING OF SPACE NORMALISED BY
MEAN RADIUS OF ANNULUS = 0.0402

SPACE NUMBER 4 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.2894

TANGENTIAL MACH NUMBER IN SPACE = 0.5625
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 2027.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.0200
DIAMETER AHEAD OF BLADE ROW (ins) = 28.14
DIAMETER AFT OF BLADE ROW (ins) = 28.53
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.7923
HUB TO TIP RATIO AFT OF BLADE ROW = 0.7751

AXIAL SPACING OF SPACE NORMALISED BY
MEAN RADIUS OF ANNULUS = 0.0162

SPACE NUMBER 5 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.2527
TANGENTIAL MACH NUMBER IN SPACE = 0.2905
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 0.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.0200
DIAMETER AHEAD OF BLADE ROW (ins) = 28.89
DIAMETER AFT OF BLADE ROW (ins) = 29.69
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.7606
HUB TO TIP RATIO AFT OF BLADE ROW = 0.7366

AXIAL SPACING OF SPACE NORMALISED BY
MEAN RADIUS OF ANNULUS = 0.0369

SPACE NUMBER 6 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.2046
TANGENTIAL MACH NUMBER IN SPACE = 0.4094
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 2027.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.0200
DIAMETER AHEAD OF BLADE ROW (ins) = 29.95
DIAMETER AFT OF BLADE ROW (ins) = 30.43
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.7311
HUB TO TIP RATIO AFT OF BLADE ROW = 0.7203

AXIAL SPACING OF SPACE NORMALISED BY
MEAN RADIUS OF ANNULUS = 0.0284

SPACE NUMBER 7 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.1907
TANGENTIAL MACH NUMBER IN SPACE = 0.2126
WHEEL RPM OF BLADE ROW AT AFT END
OF SPACE = 0.0
SPEED OF SOUND IN SPACE NORMALISED BY
EXIT SPEED OF SOUND = 1.0100
DIAMETER AHEAD OF BLADE ROW (ins) = 30.86
DIAMETER AFT OF BLADE ROW (ins) = 31.46
HUB TO TIP RATIO AHEAD OF BLADE ROW = 0.7101
HUB TO TIP RATIO AFT OF BLADE ROW = 0.6945

AXIAL SPACING OF SPACE NORMALISED BY
MEAN RADIUS OF ANNULUS = 0.0458

SPACE NUMBER 8 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE	=	0.1639
TANGENTIAL MACH NUMBER IN SPACE	=	0.2770
WHEEL RPM OF BLADE ROW AT AFT END		
OF SPACE	=	2027.0
SPEED OF SOUND IN SPACE NORMALISED BY		
EXIT SPEED OF SOUND	=	1.0100
DIAMETER AHEAD OF BLADE ROW (ins)	=	31.66
DIAMETER AFT OF BLADE ROW (ins)	=	32.05
HUB TO TIP RATIO AHEAD OF BLADE ROW	=	0.6896
HUB TO TIP RATIO AFT OF BLADE ROW	=	0.6809

AXIAL SPACING OF SPACE NORMALISED BY		
MEAN RADIUS OF ANNULUS	=	0.0247

SPACE NUMBER 9 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE	=	0.1553
TANGENTIAL MACH NUMBER IN SPACE	=	0.0944
WHEEL RPM OF BLADE ROW AT AFT END		
OF SPACE	=	0.0
SPEED OF SOUND IN SPACE NORMALISED BY		
EXIT SPEED OF SOUND	=	1.0000
DIAMETER AHEAD OF BLADE ROW (ins)	=	32.28
DIAMETER AFT OF BLADE ROW (ins)	=	32.59
HUB TO TIP RATIO AHEAD OF BLADE ROW	=	0.6750
HUB TO TIP RATIO AFT OF BLADE ROW	=	0.6657

AXIAL SPACING OF SPACE NORMALISED BY		
MEAN RADIUS OF ANNULUS	=	0.0386

SPACE NUMBER 10 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE	=	0.1441
TANGENTIAL MACH NUMBER IN SPACE	=	0.1792
WHEEL RPM OF BLADE ROW AT AFT END		
OF SPACE	=	2027.0
SPEED OF SOUND IN SPACE NORMALISED BY		
EXIT SPEED OF SOUND	=	1.0000
DIAMETER AHEAD OF BLADE ROW (ins)	=	32.63
DIAMETER AFT OF BLADE ROW (ins)	=	32.64
HUB TO TIP RATIO AHEAD OF BLADE ROW	=	0.6647
HUB TO TIP RATIO AFT OF BLADE ROW	=	0.6646

AXIAL SPACING OF SPACE NORMALISED BY		
MEAN RADIUS OF ANNULUS	=	0.0265

SPACE NUMBER 11 -@ PITCH LINE

AXIAL MACH NUMBER IN SPACE	=	0.1451
TANGENTIAL MACH NUMBER IN SPACE	=	-0.0321
WHEEL RPM OF BLADE ROW AT AFT END		
OF SPACE	=	0.0
SPEED OF SOUND IN SPACE NORMALISED BY		
EXIT SPEED OF SOUND	=	1.0000
DIAMETER AHEAD OF BLADE ROW (ins)	=	32.41
DIAMETER AFT OF BLADE ROW (ins)	=	32.00
HUB TO TIP RATIO AHEAD OF BLADE ROW	=	0.6724
HUB TO TIP RATIO AFT OF BLADE ROW	=	0.6870

AXIAL SPACING OF SPACE NORMALISED BY		
MEAN RADIUS OF ANNULUS	=	0.0265

DOWNSTREAM SPACE-@ PITCH LINE

AXIAL MACH NUMBER IN SPACE = 0.1250
TANGENTIAL MACH NUMBER IN SPACE = -0.0013

END WRITE OF INPUT PARAMETERS

*****OUTPUT FROM PROGRAM*****

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND POWER RE: 10*(-13) WATTS

TL denotes Transmission Loss

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	136.1	131.1
252.0	137.9	132.3
315.0	137.6	133.2
405.0	139.4	136.2
500.0	139.9	137.9
630.0	139.2	137.2
800.0	138.1	136.1
1000.0	136.2	134.6
1250.0	134.2	132.7
1575.0	131.4	130.4
2000.0	126.3	125.5
2520.0	117.7	117.0

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: 2*10*(-5) N/m**2

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 90.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	74.1	72.9
252.0	75.8	75.0
315.0	75.5	75.0
405.0	77.2	76.9
500.0	77.6	77.5
630.0	76.9	76.8
800.0	75.2	75.2
1000.0	72.2	72.2
1250.0	68.1	68.1
1575.0	62.0	62.0
2000.0	52.5	52.5
2520.0	42.4	42.4

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 95.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	74.3	73.1
252.0	76.1	75.3
315.0	75.8	75.3
405.0	77.6	77.3
500.0	78.1	77.9
630.0	77.5	77.4
800.0	76.2	76.1
1000.0	73.6	73.6
1250.0	70.2	70.2
1575.0	65.0	65.0
2000.0	56.0	56.0
2520.0	44.8	44.8

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 100.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	74.4	73.2
252.0	76.2	75.4
315.0	76.0	75.5
405.0	77.9	77.6
500.0	78.5	78.3
630.0	78.0	77.9
800.0	76.9	76.9
1000.0	74.7	74.7
1250.0	72.0	71.9
1575.0	67.6	67.6
2000.0	59.6	59.6
2520.0	47.8	47.8

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 105.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	74.5	73.2
252.0	76.3	75.5
315.0	76.1	75.6
405.0	78.0	77.7
500.0	78.7	78.5
630.0	78.3	78.2
800.0	77.4	77.4
1000.0	75.6	75.6
1250.0	73.3	73.3
1575.0	69.8	69.8
2000.0	62.9	62.8
2520.0	51.6	51.6

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 110.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	74.4	73.2
252.0	76.2	75.4
315.0	76.0	75.5
405.0	78.1	77.8
500.0	78.8	78.6
630.0	78.4	78.4
800.0	77.8	77.7
1000.0	76.2	76.1
1250.0	74.3	74.3
1575.0	71.4	71.4
2000.0	65.5	65.5
2520.0	55.4	55.4

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 115.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	74.2	73.0
252.0	76.0	75.2
315.0	75.9	75.3
405.0	78.0	77.7
500.0	78.7	78.5
630.0	78.4	78.3
800.0	77.9	77.8
1000.0	76.4	76.4
1250.0	74.9	74.9
1575.0	72.5	72.5
2000.0	67.5	67.5
2520.0	58.5	58.5

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 120.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	73.8	72.6
252.0	75.7	74.9
315.0	75.6	75.0
405.0	77.7	77.4
500.0	78.5	78.3
630.0	78.2	78.1
800.0	77.7	77.7
1000.0	76.4	76.4
1250.0	75.1	75.1
1575.0	73.1	73.0
2000.0	68.6	68.6
2520.0	60.4	60.4

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 125.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	73.4	72.2
252.0	75.3	74.4
315.0	75.1	74.6
405.0	77.3	77.0
500.0	78.1	77.9
630.0	77.9	77.8
800.0	77.4	77.4
1000.0	76.1	76.1
1250.0	74.9	74.9
1575.0	73.0	73.0
2000.0	68.8	68.8
2520.0	61.0	61.0

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 130.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	72.8	71.6
252.0	74.7	73.8
315.0	74.5	74.0
405.0	76.8	76.5
500.0	77.5	77.4
630.0	77.3	77.2
800.0	76.8	76.8
1000.0	75.5	75.5
1250.0	74.3	74.3
1575.0	72.4	72.3
2000.0	68.2	68.2
2520.0	60.4	60.4

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 135.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	72.1	70.9
252.0	73.9	73.1
315.0	73.8	73.3
405.0	76.0	75.7
500.0	76.8	76.6
630.0	76.5	76.4
800.0	75.9	75.9
1000.0	74.6	74.6
1250.0	73.2	73.2
1575.0	71.1	71.1
2000.0	66.6	66.6
2520.0	58.4	58.4

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 140.00

CENTER FREQUENCY,HZ	no TL dB	with TL dB
200.0	71.2	70.0
252.0	73.0	72.2
315.0	72.9	72.3
405.0	75.0	74.8
500.0	75.8	75.6
630.0	75.4	75.3
800.0	74.8	74.8
1000.0	73.3	73.3
1250.0	71.7	71.7
1575.0	69.3	69.3
2000.0	64.2	64.2
2520.0	54.9	54.9

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 145.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	70.1	68.9
252.0	71.9	71.1
315.0	71.7	71.2
405.0	73.8	73.5
500.0	74.5	74.3
630.0	74.1	74.0
800.0	73.4	73.3
1000.0	71.7	71.7
1250.0	69.7	69.7
1575.0	66.7	66.7
2000.0	60.6	60.6
2520.0	49.2	49.2

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 150.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	68.7	67.5
252.0	70.5	69.7
315.0	70.3	69.8
405.0	72.3	72.0
500.0	72.8	72.7
630.0	72.4	72.3
800.0	71.6	71.6
1000.0	69.6	69.6
1250.0	67.2	67.2
1575.0	63.3	63.3
2000.0	55.3	55.3
2520.0	38.6	38.6

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 155.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	67.1	65.9
252.0	68.9	68.0
315.0	68.6	68.1
405.0	70.4	70.1
500.0	70.8	70.6
630.0	70.4	70.3
800.0	69.4	69.4
1000.0	67.0	67.0
1250.0	64.0	64.0
1575.0	58.9	58.9
2000.0	47.2	47.2
2520.0	24.5	24.5

MIKES ON A SIDELINE,DISTANCE 400.0 FT.

EXPONENT RELATED TO CORRELATION FUNCTION = 3
OF CUT OFF RATIO BINS USED IN CALCULATION = 25

THIRD OCTAVE BAND SPL RE: $2 \times 10^{(-5)}$ N/m²

TL denotes Transmission Loss

ANGLE FROM INLET (degrees) = 160.00

CENTER FREQUENCY,Hz	no TL dB	with TL dB
200.0	65.0	63.8
252.0	66.8	66.0
315.0	66.5	66.0
405.0	67.9	67.6
500.0	68.1	67.9
630.0	67.8	67.8
800.0	66.7	66.6
1000.0	63.9	63.8
1250.0	60.0	60.0
1575.0	52.9	52.9
2000.0	29.1	29.1
2520.0	16.5	16.5

5.0 Empirical Combustor Noise Correlation

The empirical combustor noise correlation model computes the 1/3-octave SPL spectra, corrected to a 1-ft arc distance, for a 77°F standard day atmosphere. The code, called COMBUSTOR, is in standard Fortran 77. The input is in NAMELIST format, and the input parameters required are defined at the beginning of the code listing below. Output includes the input data, some calculated parameters (optional), and tabulations of the 1/3-octave SPL values at each observer angle from 10° to 170° from the inlet axis, over a range of 1/3-octave frequencies from 50 Hz to 10 kHz.

5.1 COMBUSTOR Source Code

```
C*****
C
C Program Name:  COMBUSTOR -  main program for I/O.
C
C Program Function:
C
C      Calculate the combustor noise spectra for a Single or Double
C      Annular combustor per Diversitech correlations.
C      Output source spectra are SAE 77 deg. Standard day on a 1 ft. arc.
C
C Subroutines/Functions called:
C
C      combustor_dtitrulen
C
C HPUX System routines called:
C
C      date    time
C
C*****
C
C INPUT FILE NAMELIST DEFINITIONS:
C
C Namelist /CombData/
C
C  List
C  Name      Type      Description
C  -----
C  INFO      CHAR*60 Description for predicted combustor spectra.
C  Clength   R         Combustor length, ft
C  AnnHth    R         Annulus height at combustor exit, ft
C  AreaEff   R         Combustor annulus exit area, ft**2
C  SACorDAC  CHAR*3    Combustor type selector:
C
C                      'SAC' (Single Annular Combustor)
C                      'DAC' (Double Annular Combustor)
C
C  DACType   CHAR*6    'DAC' combustor type:
C
C  config    MxFuelNz  value          fuel nozzles ignited
C  -----
C  '20-20'   40 set by program  20 outer + 20 inner
C  '20-10'   40 set by user     20 outer + 10 alternating inner  (101010...)
C  '20-10'   30 set by user     20 outer + 10 inner
C  '20-2^5'  40 set by program  20 outer + 10 alternating pairs of inner (110011...)
C  '20-0'    30/40 set by user  20 outer + 0 inner
C
C  MxFuelNz  INTG      Total number of DAC combustor fuel nozzles for
```

```

C          DACType(s) = '20-10', or '20-0' only. (use 30 or 40)
C  NumFuel  INTG    Number of 'SAC' fuel nozzles ignited
C  DIAHYD   R       Core Nozzle Exit Plane Hydraulic diameter, ft
C  DIAEFF   R       Core Nozzle Exit Plane Effective diameter, ft
C  W3       R       Weight Flow at HP Compressor discharge, lbm/sec
C  T3       R       Temperature at HP Compressor discharge, degrees R
C  P3       R       Pressure at HP Compressor discharge, psia
C  T4       R       Temperature HP Turbine Inlet, degrees R
C  P4       R       Pressure at HP Turbine Inlet, psia
C  T8       R       Temperature at Primary Nozzle (core) throat, degrees R
C  P8       R       Pressure at Primary Nozzle (core) throat, psia
C  IDIAG    INTG    Diagnostic output flag: 0 = no, 1 = yes.
C
C

```

C SCREEN OR OUTPUT FILE CONTENTS:

```

C  Name      Description
C  -----
C  INFO      Description for predicted combustor spectra.
C  IALPHA     Data is predicted using the SAE (ARP866A) model.
C  TAMB       Ambient temperature, degrees F
C  PAMB       Ambient pressure, inches Hg (mercury)
C  RELHUM     Ambient Percent relative humidity
C  IARCSL     Data is predicted on an 'ARC'
C  DIST       Data is predicted on a 1 ft. arc.
C  SACorDAC   Combustor type: 'SAC' or 'DAC'
C  DACTYPE    Configuration of the DAC combustor
C  Clength    Combustor length, ft
C  AnnHth     Annulus height at combustor exit, ft
C  AreaEff    Combustor annulus exit area, ft**2
C  NumFuel    Number of fuel nozzles ignited (SAC or DAC)
C  MxFuelNz   Total number of DAC combustor fuel nozzles
C  DIAHYD     Combustor hydraulic diameter, ft
C  DIAEFF     Combustor effective diameter, ft
C  T3         Temperature at HP Compressor discharge, degrees R
C  P3         Pressure at HP Compressor discharge, psia
C  T4         Temperature HP Turbine Inlet, degrees R
C  P4         Pressure at HP Turbine Inlet, psia
C  T8         Temperature at Primary Nozzle (core) throat, degrees R
C  P8         Pressure at Primary Nozzle (core) throat, psia
C  SPL        Total Summed SPL(angle,freq) Tabular column spectra
C             for each acoustic angle (10 - 170 degrees).
C

```

C If IDIAG flag = 1 then these additional parameters are output:

```

C  FFC        Correlation flow function correction term
C  FFT        Correlation turbine loss correction term
C  XK1        Turbine transmission loss parameter
C  SPLFFC     SPL value for FFC
C  SPLTL      SPL value for FFT
C
C  If SAC: CP      Correlation parameter for 63, 160, 630 Hz.
C                OASPK63   Peak OASPL for 63 Hz band.
C                OASPK160  Peak OASPL for 160 Hz band.
C                OASPK630  Peak OASPL for 630 Hz band.
C
C  If DAC: CP160   Correlation Parameter value for 160 Hz.
C                CP500    Correlation Parameter value for 500 Hz.
C                OASPK160  Peak OASPL for 160 Hz band.
C                OASPK500  Peak OASPL for 500 Hz band.

```

```

C
C*****
C
C      PROGRAM COMBUSTOR
C
C      PARAMETER( NANG = 17 )
C      PARAMETER( NFREQ = 24 )
C
C      CHARACTER *60 INFO
C
C      CHARACTER *3  SACORDAC  /' '/
C      CHARACTER *6  DACTYPE   /' '/
C      CHARACTER *255 NLIN     /' '/
C      CHARACTER *255 TABOUT   /' '/
C      CHARACTER *1   IANS      /' '/
C      CHARACTER *9   ADATE     /' '/
C      CHARACTER *8   ATIME     /' '/
C      CHARACTER *16  PROGNAM   /'COMBUSTOR'/
C      CHARACTER *4   PROGVER   /'1.2'/
C
C      DIMENSION COMBSPL(NANG,NFREQ)
C      DIMENSION ANG      (NANG)
C      DIMENSION FREQS    (NFREQ)
C
C      DATA ANG  /10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, 80.0,
& 90.0, 100.0, 110.0, 120.0, 130.0, 140.0, 150.0, 160.0, 170.0/
C
C      DATA FREQS/ 50.0, 63.0, 80.0, 100.0, 125.0,
& 160.0, 200.0, 250.0, 315.0, 400.0,
& 500.0, 630.0, 800.0, 1000.0, 1250.0,
& 1600.0, 2000.0, 2500.0, 3150.0, 4000.0,
& 5000.0, 6300.0, 8000.0,10000.0/
C
C *** Namelist Input
C
C      NAMELIST /COMBDATA/
& INFO, SACORDAC, DACTYPE, CLENGTH, ANNHTH,
& AREAEFF, NUMFUEL, MXFUELNZ, DIAHYD, DIAEFF,
& W3, T3, P3, T4, P4, T8, P8, IDIAG
C
C *** Output Program description
C
C      CALL DATE (ADATE)
C      CALL TIME (ATIME)
C      WRITE(06,*) ' '
C      WRITE(06,*) ' PROGRAM: ',PROGNAM(1:itrulen(PROGNAM)),
& ' Version ',PROGVER,' ',ADATE,' ',ATIME
C      WRITE(06,*) ' '
C      WRITE(06,*) 'This program predicts combustor noise spectra using'
C      WRITE(06,*) 'correlation procedures developed by Diversitech'
C      WRITE(06,*) 'while under subcontract to GEAE.'
C      WRITE(06,*) 'The combustor data were provided by GEAE.'
C      WRITE(06,*) ' '
C      WRITE(06,'(1X$,A23)') 'enter <cr> to continue '
C      READ(05,'(A)') IANS
C      WRITE(06,*) ' '
C
C *** Get Namelist input filename.
C
C      NLIN = ' '

```

```

DO WHILE (ITRLEN(NLIN) .LE. 0)
  WRITE(06,*)'Enter Namelist input filename:'
  READ(05,'(A)') NLIN
ENDDO
OPEN(UNIT=7, FILE=NLIN, STATUS='UNKNOWN',ERR=1000, IOSTAT=II)
C
C *** Write results to the screen or tabular output file?
C
  IANS = ' '
  DO WHILE (ITRLEN(IANS) .LE. 0)
    WRITE (06,'(1X$,A49)')
    & 'Output Tabular results to Screen or File (S/F)?: '
    READ(05,'(A)') IANS
    IF(IANS .EQ. 'S' .OR. IANS .EQ. 's') THEN
      IFC = 6
    ELSEIF(IANS .EQ. 'F' .OR. IANS .EQ. 'f') THEN
      IFC = 8
      TABOUT = ' '
      DO WHILE (ITRLEN(TABOUT) .LE. 0)
        WRITE(06,*)'Enter Tabular data output filename:'
        READ(05,'(A)') TABOUT
      ENDDO
      OPEN (UNIT=IFC, FILE=TABOUT, STATUS='UNKNOWN',
&      ERR=1000, IOSTAT=II)
    ELSE
      IANS = ' '
    ENDIF
  ENDDO
C
C *** Read a case from the namelist input file.
C
100 CONTINUE
READ(UNIT=7, NML=COMBDATA, END=9999)
C
C *** Check some input parameters.
C
  IF(SACORDAC .NE. 'SAC' .AND. SACORDAC .NE. 'DAC') THEN
    WRITE(06,*)
    & '*ERROR: Invalid SACORDAC entry. Must be SAC or DAC'
    CALL EXIT
  ENDIF
  IF(SACORDAC .EQ. 'DAC') THEN
    IOK = 0
    IF(DACTYPE .EQ. '20-20' ) THEN
      IOK = 1
      MXFUELNZ = 40
    ELSEIF(DACTYPE .EQ. '20-10' ) THEN
      IOK = 1
      IF(MXFUELNZ .NE. 30 .AND. MXFUELNZ .NE. 40) THEN
        WRITE(06,*)
        & '*ERROR: Invalid MXFUELNZ value for DACTYPE = ',DACTYPE
        WRITE(06,*)
        & '          Current value = ',MXFUELNZ
        WRITE(06,*)
        & '          Set correct value in input file and rerun.'
        CALL EXIT
      ENDIF
    ELSEIF(DACTYPE .EQ. '20-2^5') THEN
      MXFUELNZ = 40
      IOK = 1

```

```

        ELSEIF(DACTYPE .EQ. '20-0' ) THEN
            IOK = 1
            IF(MXFUELNZ .NE. 30 .AND. MXFUELNZ .NE. 40) THEN
                WRITE(06,*)
&                '*ERROR: Invalid MXFUELNZ value for DACTYPE = ',DACTYPE
                WRITE(06,*)
&                '          Current value = ',MXFUELNZ
                WRITE(06,*)
&                '          Set correct value in input file and rerun.'
                CALL EXIT
            ENDIF
        ENDIF
    ENDIF
    IF(IOK .EQ. 0) THEN
        WRITE(06,*) '*ERROR: Invalid DACTYPE entry'
        WRITE(06,*) '          Must be 20-20, 20-10, 20-2^5, or 20-0'
        CALL EXIT
    ENDIF
ENDIF

C
C *** Call COMBUSTOR_DT to calculate 77 deg. F SAE Std day, 1 ft. Arc
C *** Combustor spectra.
C
    CALL COMBUSTOR_DT (CLENGTH,ANNHTH,AREAEFF,NUMFUEL,MXFUELNZ,
&                    DIAHYD,DIAEFF,W3,T3,P3,T4,P4,T8,P8,
&                    SACORDAC,DACTYPE,COMBSPL,CP,CP160,CP500,
&                    FFC,FFT,XK1,SPLFFC,SPLTL,OASPK63,
&                    OASPK160,OASPK630,OASPK500)

C
C *** Output the Tabular Combustor SPL(angle, freq) & OASPL(angle) data
C
    IF(IFC .NE. 06) THEN
        WRITE(IFC,*)'PROGRAM: ',PROGNAM,' Version ',PROGVER,
&                ' ',ADATE,' ',ATIME

    ENDIF

C
    WRITE(IFC,*)' '
    WRITE(IFC,*)'GEAE/ADT Combustor Noise Prediction'
    WRITE(IFC,*)' - Modified Diversitech Correlation'
    WRITE(IFC,*)' '
    WRITE(IFC,*) 'INFO          = ',INFO(1:ITRULEN(INFO))
    WRITE(IFC,*) 'IALPHA        = SAE77'
    WRITE(IFC,*) 'TAMB          = 77 Deg. F'
    WRITE(IFC,*) 'PAMB          = 29.92 In. Hg'
    WRITE(IFC,*) 'RELHUM        = 70%'
    WRITE(IFC,*) 'DIST          = 1 Ft.'
    WRITE(IFC,*) 'IARCSL        = ARC'
    WRITE(IFC,*) 'SACORDAC      = ',SACORDAC
    IF(SACORDAC .eq. 'DAC') WRITE(IFC,*) 'DACTYPE    = ',DACTYPE
    WRITE(IFC,*) 'CLENGTH       = ',CLENGTH
    WRITE(IFC,*) 'ANNHTH        = ',ANNHTH
    WRITE(IFC,*) 'AREAEFF       = ',AREAEFF
    WRITE(IFC,*) 'NUMFUEL       = ',NUMFUEL
    IF(SACORDAC .eq. 'DAC') WRITE(IFC,*) 'MXFUELNZ    = ',MXFUELNZ
    WRITE(IFC,*) 'DIAHYD        = ',DIAHYD
    WRITE(IFC,*) 'DIAEFF        = ',DIAEFF
    WRITE(IFC,*) 'W3            = ',W3
    WRITE(IFC,*) 'T3            = ',T3
    WRITE(IFC,*) 'P3            = ',P3
    WRITE(IFC,*) 'T4            = ',T4
    WRITE(IFC,*) 'P4            = ',P4

```

```

WRITE(IFC,*) 'T8          = ',T8
WRITE(IFC,*) 'P8          = ',P8
IF(IDIAG .EQ. 1) THEN
  WRITE(IFC,*) 'FFC        = ',FFC
  WRITE(IFC,*) 'FFT        = ',FFT
  WRITE(IFC,*) 'SPLFFC     = ',SPLFFC
  WRITE(IFC,*) 'SPLTL      = ',SPLTL
  WRITE(IFC,*) 'XK1        = ',XK1
ENDIF
IF(SACorDAC .eq. 'SAC') THEN
  IF(IDIAG .EQ. 1) THEN
    WRITE(IFC,*) 'CP        = ',CP
    WRITE(IFC,*) 'OASPK63   = ',OASPK63
    WRITE(IFC,*) 'OASPK160  = ',OASPK160
    WRITE(IFC,*) 'OASPK630  = ',OASPK630
  ENDIF
  WRITE(IFC,*) ' '
  WRITE(IFC,*) 'Single Annular Combustor Spectra'
ELSEIF(SACorDAC .eq. 'DAC') THEN
  IF(IDIAG .EQ. 1) THEN
    WRITE(IFC,*) 'CP160     = ',CP160
    WRITE(IFC,*) 'CP500     = ',CP500
    WRITE(IFC,*) 'OASPK160  = ',OASPK160
    WRITE(IFC,*) 'OASPK500  = ',OASPK500
  ENDIF
  WRITE(IFC,*) ' '
  WRITE(IFC,*) 'Double Annular Combustor Spectra'
ENDIF
C
C *** Print out angles 10 thru 90.
C
  WRITE(IFC,*) ' '
  WRITE(IFC,*(24X,A32)) 'Acoustic Angles from Inlet, deg.'
  WRITE(IFC,*(8X,9F7.1)) (ANG(II),II = 1,9)
  WRITE(IFC,*) ' Freq.'
  DO JJ = 1,NFREQ
    WRITE(IFC,*(1X,F7.1,9F7.2))
    &   FREQS(JJ),(COMBSPL(II,JJ),II = 1,9)
  ENDDO
  WRITE(IFC,*) ' '
C
C *** Print out angles 100 thru 170.
C
  WRITE(IFC,*) ' '
  WRITE(IFC,*(24X,A32)) 'Acoustic Angles from Inlet, deg.'
  WRITE(IFC,*(8X,8F7.1)) (ANG(II),II = 10,17)
  WRITE(IFC,*) ' Freq.'
  DO JJ = 1,NFREQ
    WRITE(IFC,*(1X,F7.1,8F7.2))
    &   FREQS(JJ),(COMBSPL(II,JJ),II = 10,17)
  ENDDO
  WRITE(IFC,*) ' '
C
C *** Loop back and read another input case.
C
  GOTO 100
C
1000 PRINT *,II
9999 IF(IFC .NE. 6) CLOSE (IFC)
C

```

```

        STOP
        END
C
C
C>>*****
C
C Subroutine Name:  COMBUSTOR_DT
C (GEAE modified version of original Diversitech program: Combnoise)
C
C Subroutine Function:
C
C       Calculate the combustor noise spectra for a Single or Double
C       Annular combustor per Diversitech correlations.
C
C Comments:  The engine spectral data provided to Diversitech by GEAE
C            were corrected to 150 ft. arc, 77 deg. F Standard day
C            conditions using the SAE ARP866A atmospheric attenuation
C            model. The combustor component spectra were extracted
C            from the total engine spectra by Diversitech under the
C            guidance of GEAE Acoustics.  Diversitech then correlated
C            the combustor spectral data and developed the prediction
C            equations and constants. rsc - 06apr98
C
C Routines called:
C
C       none
C
C*****
C
C   Input Arguments:
C
C   Name          Type      Description
C   -----
C   CLENGTH       R         COMBUSTOR LENGTH, FT
C   ANNHETH       R         ANNULUS HEIGHT AT COMBUSTOR EXIT, FT
C   AREAIEFF      R         COMBUSTOR ANNULUS EXIT AREA, FT**2
C   NUMFUEL       INTG      NUMBER OF 'SAC' FUEL NOZZLES IGNITED
C   MXFUELNZ      INTG      TOTAL NUMBER OF 'DAC' COMBUSTOR FUEL NOZZLES
C   DIAHYD        R         COMBUSTOR HYDRALIC DIAMETER, FT
C   DIAEFF        R         COMBUSTOR EFFECTIVE DIAMETER, FT
C   W3            R         WEIGHT FLOW AT HP COMPRESSOR DISCHARGE, LBM/SEC
C   T3            R         TEMPERATURE AT HP COMPRESSOR DISCHARGE, DEGREES R
C   P3            R         PRESSURE AT HP COMPRESSOR DISCHARGE, PSIA
C   T4            R         TEMPERATURE HP TURBINE INLET, DEGREES R
C   P4            R         PRESSURE AT HP TURBINE INLET, PSIA
C   T8            R         TEMPERATURE AT PRIMARY NOZZLE (CORE) THROAT, DEGREES R
C   P8            R         PRESSURE AT PRIMARY NOZZLE (CORE) THROAT, PSIA
C   SACORDAC      CHAR*3    COMBUSTOR TYPE SELECTOR:
C                           'SAC' = Single Annular Combustor
C                           'DAC' = Double Annular Combustor
C   DACTYPE       CHAR*6    DAC TYPE COMBUSTER TYPE:
C                           '20-20', '20-10', '20-2^5', or '20-0'
C   Output Arguments:
C
C   NUMFUEL       INTG      NUMBER OF FUEL NOZZLES IGNITED (SAC or DAC)
C   COMBSPL       R         PREDICTED COMBUSTOR SPECTRA AT 1 FT. ARC, 77 DEG. F
C                           STANDARD DAY FOR
C                           24 FREQUENCIES: 50-10KHZ
C                           17 ACOUSTIC ANGLES: 10 - 170 DEGREES
C   CP            R         'SAC' CORRELATION PARAMETER USED TO CALC PEAK ANGLE OASPL

```

```

C          AT 63, 160, 630 Hz.
C   CP160      R      'DAC' CORRELATION PARAMETER USED TO CALC PEAK ANGLE OASPL
C          AT 160 Hz
C   CP500      R      'DAC' CORRELATION PARAMETER USED TO CALC PEAK ANGLE OASPL
C          AT 500 Hz
C   FFC        R      Correlation flow function correction term
C   FFT        R      Correlation turbine loss correction term
C   XK1        R      Turbine transmission loss parameter
C   SPLFFC     R      SPL value for FFC
C   SPLTL      R      SPL value for FFT
C   OASPK63    R      Peak OASPL for 63 Hz band (SAC only)
C   OASPK160   R      Peak OASPL for 160 Hz band (SAC and DAC)
C   OASPK630   R      Peak OASPL for 630 Hz band (SAC only)
C   OASPK500   R      Peak OASPL for 500 Hz band (DAC only)
C
C>>*****
C
C      SUBROUTINE COMBUSTOR_DT (CLENGTH,ANNHTH,AREAEFF,NUMFUEL,MXFUELNZ,
C      & DIAHYD,DIAEFF,W3,T3,P3,T4,P4,T8,P8,
C      & SACORDAC,DACTYPE,COMBSPL,CP,CP160,CP500,
C      & FFC,FFT,XK1,SPLFFC,SPLTL,OASPK63,
C      & OASPK160,OASPK630,OASPK500)
C
C      PARAMETER( NUMALPHA = 17 )
C      PARAMETER( NUMFREQS = 24 )
C
C      CHARACTER *(*) SACORDAC
C      CHARACTER *(*) DACTYPE
C
C      DIMENSION XNRM63 ( NUMALPHA ), OASPL63 ( NUMALPHA ),
C      & XNRM160( NUMALPHA ), OASPL160( NUMALPHA ),
C      & XNRM500( NUMALPHA ), OASPL500( NUMALPHA ),
C      & XNRM630( NUMALPHA ), OASPL630( NUMALPHA ),
C      & SPL63 ( NUMALPHA, NUMFREQS ),
C      & SPL160( NUMALPHA, NUMFREQS ),
C      & SPL500( NUMALPHA, NUMFREQS ),
C      & SPL630( NUMALPHA, NUMFREQS )
C
C      DIMENSION COMBSPL (NUMALPHA,NUMFREQS)
C      DIMENSION ANG      (NUMALPHA)
C      DIMENSION FREQS    (NUMFREQS)
C      DIMENSION ATMOSABS (NUMFREQS)
C
C      DATA ALPK63  /150.0/
C      DATA ALPK160 /130.0/
C      DATA ALPK500 /130.0/
C      DATA ALPK630 /130.0/
C      DATA FPK63   /63.0/
C      DATA FPK160  /160.0/
C      DATA FPK500  /500.0/
C      DATA FPK630  /630.0/
C      DATA MINS63 , MAXS63  /1,5/
C      DATA MINS160, MAXS160 /2,9/
C      DATA MINS630, MAXS630 /6,14/
C      DATA MIND160, MAXD160 /1,9/
C      DATA MIND500, MAXD500 /5,14/
C      DATA RADIAL  /150.0/
C      DATA PAMB_PSIA /14.696/      !29.92 in. Hg
C      DATA TAMB     /77.0/
C      DATA RELHUM    /70.0/
C

```

```

DATA ANG /10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 70.0, 80.0,
& 90.0, 100.0, 110.0, 120.0, 130.0, 140.0, 150.0, 160.0, 170.0/
C
DATA FREQS/ 50.0, 63.0, 80.0, 100.0, 125.0,
& 160.0, 200.0, 250.0, 315.0, 400.0,
& 500.0, 630.0, 800.0, 1000.0, 1250.0,
& 1600.0, 2000.0, 2500.0, 3150.0, 4000.0,
& 5000.0, 6300.0, 8000.0, 10000.0/
C
C *** Set ARP866A model atmospheric absorption values for each
C *** frequency (50-10khz) at 1000 ft for 77 deg.f standard day.
C
DATA ATMOSABS / 0.08700, 0.10966, 0.13929, 0.17419,
& 0.21785, 0.27906, 0.34913, 0.43690,
& 0.55128, 0.70138, 0.87871, 1.11046,
& 1.41565, 1.77779, 2.23524, 2.88468,
& 3.63993, 4.60376, 5.88995, 7.62918,
& 8.58712, 11.07580, 14.94318, 20.32406/
C
C *** Initialize subroutine and output data.
C
FFC = 0.00
FFT = 0.00
XK1 = 0.00
SPLFFC = 0.00
SPLTL = 0.00
C
OASPK63 = 0.00
OASPK160 = 0.00
OASPK500 = 0.00
OASPK630 = 0.00
C
DO I = 1, NUMALPHA
XNRM63 (I) = 0.00
XNRM160 (I) = 0.00
XNRM500 (I) = 0.00
XNRM630 (I) = 0.00
OASPL63 (I) = 0.00
OASPL160(I) = 0.00
OASPL500(I) = 0.00
OASPL630(I) = 0.00
DO J = 1, NUMFREQS
SPL63 (I,J) = 0.00
SPL160(I,J) = 0.00
SPL500(I,J) = 0.00
SPL630(I,J) = 0.00
COMBSPL(I,J) = 0.0
ENDDO
ENDDO
C
C *** Calculate the correlation flow function correction term FFC
C *** along with SPL(FFC).
C *** If DAC: set number of inner fuel nozzles ignited (NUM_INNER) based on DACTYPE.
C *** set Total number of ignited fuel nozzles (NUMFUEL)
C
IF(SACORDAC .EQ. 'SAC') THEN
FFC = W3*SQRT((T4-T3))/(P3*FLOAT(NUMFUEL)*AREAEFF**2)
ELSEIF(SACORDAC .EQ. 'DAC') THEN
IF (DACTYPE .EQ. '20-20 ') THEN
NUM_INNER = 20

```

```

        NUMFUEL = 40
        XK = 0.250
    ELSEIF (DACTYPE .EQ. '20-10 ') THEN
        NUM_INNER = 10
        NUMFUEL = 30
        XK = 0.250
    ELSEIF(DACTYPE .EQ. '20-2^5') THEN
        NUM_INNER = 10
        NUMFUEL = 30
        XK = 0.200
    ELSEIF(DACTYPE .EQ. '20-0') THEN
        NUM_INNER = 0
        NUMFUEL = 20
        XK = 0.000
    ENDIF
    FFC = W3*SQRT((T4-T3))/(P3*SQRT(FLOAT(20+NUM_INNER))*AREAEFF**2)
    ENDIF
    SPLFFC = 20.0D0 * LOG10(FFC)
C
C *** Calculate the correlation turbine loss correction term FFT
C *** along with SPL(FFT).

    PI = ACOS(-1.0D0)
    FFT = (P4/P8)*SQRT(T8/T4)
    XK1 = ((1.0D0+FFT)**2)/(4.0D0*CLENGTH*FFT/(PI*ANNHTH))
    SPLTL = 20.0D0*LOG10(XK1)
C
C *** Calculate the cycle and geometry dependent term CP used
C *** in the OASPL(PEAK ANGLE) correlation.
C
    IF(SACORDAC .EQ. 'SAC') THEN
C
        CP = ((W3*SQRT(T3))/P3)*((T4-T3)/T4)*
&          (P3/PAMB_PSIA)*(DIAHYD/DIAEFF)**0.500
C
    ELSEIF(SACORDAC .EQ. 'DAC') THEN
C
        CP160 = ((W3*SQRT(T3))/P3)*((T4-T3)/T4)*
&              (P3/PAMB_PSIA)*(DIAHYD/DIAEFF)**2.000
        CP500 = ((W3*SQRT(T3))/P3)*((T4-T3)/T4)*
&              (P3/PAMB_PSIA)*(DIAHYD/DIAEFF)**1.200
    ENDIF
C
C *** Calculate OASPL(PEAK ANGLE) correlation.
C
    IF (SACORDAC .EQ. 'SAC') THEN
C
        HSAC63 = 14.2560 * LOG(CP) + 76.450
        HSAC160 = 3.3150 * LOG(CP) + 108.500
        HSAC630 = 6.9380 * LOG(CP) + 106.380
        OASPK63 = -20.00 * LOG10(RADIAL) + HSAC63 *
&                (30.00/FLOAT(NUMFUEL))**(-0.225)
        OASPK160 = -20.00 * LOG10(RADIAL) + HSAC160 *
&                (30.00 / FLOAT(NUMFUEL))**0.050
        OASPK630 = -20.00 * LOG10(RADIAL) + HSAC630 *
&                (30.00 / FLOAT(NUMFUEL))**0.020
C
    ELSEIF(SACORDAC .EQ. 'DAC') THEN
C
        HDAC160 = 2.69310 * LOG(CP160) + 110.440
        HDAC500 = 2.99170 * LOG(CP500) + 110.620

```

```

C      IF (NUM_INNER .EQ. 20) THEN          !20 inner nozzles ignited
C
      OASPK160 = 1.200 * (-20.00 * LOG10(RADIAL) + HDAC160 *
&      ((20.0 + FLOAT(NUM_INNER)) /
&      FLOAT(MXFUELNZ))**(-XK)) *
&      ( 30.00 / (20.0 + FLOAT(NUM_INNER)))**0.020)
      OASPK500 = -20.00 * LOG10(RADIAL) + HDAC500 *
&      ((20.0 + FLOAT(NUM_INNER)) /
&      FLOAT(MXFUELNZ))**(-XK)) *
&      ( 30.00 / (20.0 + FLOAT(NUM_INNER)))**0.180
C
      ELSEIF (NUM_INNER .EQ. 10) THEN !10 inner nozzles ignited
C
      OASPK160 = 0.9800 * (-20.00 * LOG10(RADIAL) + HDAC160 *
&      ((20.0 + FLOAT(NUM_INNER)) /
&      FLOAT(MXFUELNZ))**(-XK)) *
&      ( 30.00 / (20.0 + FLOAT(NUM_INNER)))**0.020)
      OASPK500 = 0.9000 * (-20.00 * LOG10(RADIAL) + HDAC500 *
&      ((20.0 + FLOAT(NUM_INNER)) /
&      FLOAT(MXFUELNZ))**(-XK)) *
&      ( 30.00 / (20.0 + FLOAT(NUM_INNER)))**0.180)
C
      ELSEIF (NUM_INNER .EQ. 0) THEN !0 inner nozzles ignited
C
      OASPK160 = 1.1000 * (-20.00 * LOG10(RADIAL) + HDAC160 *
&      ((20.0 + FLOAT(NUM_INNER)) /
&      FLOAT(MXFUELNZ))**(-XK)) *
&      ( 30.00 / (20.0 + FLOAT(NUM_INNER)))**0.020)
      OASPK500 = 0.9800 * (-20.00 * LOG10(RADIAL) + HDAC500 *
&      ((20.0 + FLOAT(NUM_INNER)) /
&      FLOAT(MXFUELNZ))**(-XK)) *
&      ( 30.00 / (20.0 + FLOAT(NUM_INNER)))**0.180)
      END IF
      END IF
C
C *** Calculate the correlation norm value of OASPL for a
C      SAC or DAC combustor configuration.
C
      DO II = 1, NUMALPHA
      IF(SACORDAC .EQ. 'SAC') THEN
      ALPHN63 = ANG(II)/ALPK63
      ALPHN160 = ANG(II)/ALPK160
      ALPHN630 = ANG(II)/ALPK630
      XNRM63 (II) = -67.8*(ALPHN63)**2+141.7*(ALPHN63)-66.84
      XNRM160(II) = -26.019*(ALPHN160)**3-5.2974*(ALPHN160)**2
&      + 93.433*(ALPHN160)- 61.751
      XNRM630(II) = -156.5*(ALPHN630)**2+322.340*(ALPHN630)-164.89
      ELSEIF(SACORDAC .EQ. 'DAC') THEN
      ALPHN160 = ANG(II)/ALPK160
      ALPHN500 = ANG(II)/ALPK500
      XNRM160(II) = -116.95*(ALPHN160)**2+235.23*(ALPHN160)-120.65
      XNRM500(II) = -137.59*(ALPHN500)**2+283.40*(ALPHN500)-147.73
      ENDIF
      ENDDO
C
C *** Calculate the peak frequencies OASPL(ANGLE) for a SAC or DAC
C
      DO II = 1, NUMALPHA
      IF(SACORDAC .EQ. 'SAC') THEN

```

```

        OASPL63 (II) = XNRM63 (II)+OASPK63 +0.40*(SPLFFC-SPLTL)
        OASPL160(II) = XNRM160(II)+OASPK160+0.10*(SPLFFC-SPLTL)
        OASPL630(II) = XNRM630(II)+OASPK630+0.30*(SPLFFC-SPLTL)
    ELSEIF(SACORDAC .EQ. 'DAC') THEN
        OASPL160(II) = XNRM160(II)+OASPK160+0.45*(SPLFFC-SPLTL)
        OASPL500(II) = XNRM500(II)+OASPK500-0.10*(SPLFFC-SPLTL)
    ENDIF
ENDDO

C
C *** Data correlations are based on 150 ft. arc data corrected to
C *** 77 deg.f standard day using the SAE ARP866A atmospheric attenuation
C *** model.
C
C *** Calculate the 1 ft. arc source SPL(ANGLE, FREQ) for a SAC or DAC
C *** combustor configuration.
C
    SD = 20.0*LOG10(150.0/1.0) !spherical divergence component
    DO II = 1, NUMALPHA
        IF(SACORDAC .EQ. 'SAC') THEN
            DO JJ = MINS63, MAXS63
                FR63 = FREQS(JJ)/FPK63
                SPL63(II,JJ) = OASPL63(II)- 145.610*(FR63)**2+295.460*
&                                (FR63)-152.720+ATMOSABS(JJ)*149.0/1000.0
&                                + SD
                IF(SPL63(II,JJ) .GT. 0.0) THEN
                    COMBSPL(II,JJ) = SPL63(II,JJ)
                ENDIF
            ENDDO
            DO KK = MINS160, MAXS160
                FR160 = FREQS(KK)/FPK160
                SPL160(II,KK) = OASPL160(II)-163.340*(FR160)**2+331.330*
&                                (FR160)-170.070+ATMOSABS(KK)*149.0/1000.0
&                                + SD
                IF(SPL160(II,KK) .GT. 0.0) THEN
                    IF(COMBSPL(II,KK) .GT. 0.0) THEN
                        COMBSPL(II,KK) = 10.0*LOG10(10.0**((SPL160(II,KK)/10.0)+
&                                10.0**((COMBSPL(II,KK)/10.0))
                        ELSE
                            COMBSPL(II,KK) = SPL160(II,KK)
                        ENDIF
                    ENDIF
                ENDIF
            ENDDO
            DO LL = MINS630, MAXS630
                FR630 = FREQS(LL)/FPK630
                SPL630(II,LL) = OASPL630(II)-142.310*(FR630)**2+286.440*
&                                (FR630)-147.500+ATMOSABS(LL)*149.0/1000.0
&                                + SD
                IF(SPL630(II,LL) .GT. 0.0) THEN
                    IF(COMBSPL(II,LL) .GT. 0.0) THEN
                        COMBSPL(II,LL) = 10.0*LOG10(10.0**((SPL630(II,LL)/10.0)+
&                                10.0**((COMBSPL(II,LL)/10.0))
                        ELSE
                            COMBSPL(II,LL) = SPL630(II,LL)
                        ENDIF
                    ENDIF
                ENDIF
            ENDDO
        ELSEIF(SACORDAC .EQ. 'DAC') THEN
            DO KK = MIND160, MAXD160

```

```

        FR160 = FREQS(KK)/FPK160
        SPL160(II,KK) = OASPL160(II)-143.000*(FR160)**2+280.040*
&                (FR160)-143.070+ATMOSABS(KK)*149.0/1000.0
&                + SD
        IF(SPL160(II,KK) .GT. 0.0) THEN
            COMBSPL(II,KK) = SPL160(II,KK)
        ENDIF
    ENDDO
    DO LL = MIND500, MAXD500
        FR500 = FREQS(LL)/FPK500
        SPL500(II,LL) = OASPL500(II)-135.810*(FR500)**2+268.990*
&                (FR500)-137.210+ATMOSABS(LL)*149.0/1000.0
&                + SD
        IF(SPL500(II,LL) .GT. 0.0) THEN
            IF(COMBSPL(II,LL) .GT. 0.0) THEN
                COMBSPL(II,LL) = 10.0*LOG10(10.0**((SPL500(II,LL)/10.0)+
&                10.0**((COMBSPL(II,LL)/10.0))
            ELSE
                COMBSPL(II,LL) = SPL500(II,LL)
            ENDIF
        ENDIF
    ENDDO
ENDIF
ENDDO
C
9999 RETURN
END

c
c***** function itrulen *****
c
c        function itrulen(string)
c
c        description      - A function to determine the true length of a string
c
c        varaible glossary
c
c        character variables
c        string          - string for which the length is needed - input
c
c        integer variables
c        i                - loop counter - internal
c        maxlen          - the maximum string length return from the len function &
c                        is equal to the defined character length in the calling
c                        program - internal
c        itrulen         - the length of the string - output
c
c        define variables
c
c        character*(*) string
c        integer i,maxlen,itrulen
c
c        Set length equal to maxlen then work backward until the first
c        printable ascii character is found (in the ascii value range 33 to 126).
c
c        maxlen=len(string)
c        i=maxlen+1
c        loop = 1
c        do while (loop .eq. 1 .and. i .gt. 0)
c            i=i-1
c            ival = ichar(string(i:i))

```

```

        if(ival .gt. 32 .and. ival .lt. 127) loop = 0
    end do
c
c return the true length
c
    itrulen=i
    return
end

```

5.2 COMBUSTOR Sample Input File

```

$COMBDATA
INFO      ='Sample SAC Combustor',
SACorDAC  ='SAC',
Clength   =0.4583,
AnnHth    =0.2083,
AreaEff    =1.499,
NumFuel    =20,
DIAHYD    =13.84,
DIAEFF    =34.59,
W3         =86.707,
T3         =1036.5,
P3         =114.39,
T4         =1889.4,
P4         =109.07,
T8         =1084.0,
P8         =14.847
IDIAG     =1
$
$COMBDATA
INFO      ='Sample DAC Combustor',
SACorDAC  ='DAC',
DACType   ='20-10',
MXFUELNZ  = 40,
Clength   =0.492,
AnnHth    =0.250,
AreaEff    =2.430,
DIAHYD    =13.4,
DIAEFF    =39.9,
W3         =113.9,
T3         =1189.9,
P3         =173.1,
T4         =2260.,
P4         =162.9,
T8         =1205.6,
P8         =15.1,
$

```

5.3 COMBUSTOR Sample Output File

```

PROGRAM: COMBUSTOR          Version 1.2  17-May-99 10:01:16

GEAE/ADT Combustor Noise Prediction
- Modified Diversitech Correlation

INFO      = Sample SAC Combustor
IALPHA    = SAE77
TAMB      = 77 Deg. F
PAMB      = 29.92 In. Hg
RELHUM    = 70%
DIST      = 1 Ft.

```

```

IARCSL      = ARC
SACORDAC    = SAC
CLENGTH     = .4583
ANNHTH      = .2083
AERAEFF     = 1.499
NUMFUEL     = 20
DIAHYD      = 13.84
DIAEFF      = 34.59
W3          = 86.707
T3          = 1036.5
P3          = 114.39
T4          = 1889.4
P4          = 109.07
T8          = 1084.0
P8          = 14.847
FFC         = .4925858
FFT         = 5.56441
SPLFFC      = -6.15036
SPLTL       = 8.83203
XK1         = 2.76441
CP          = 54.23838
OASPK63     = 78.22828
OASPK160    = 80.70947
OASPK630    = 91.65606

```

Single Annular Combustor Spectra

	Acoustic Angles from Inlet, deg.								
	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0
Freq.									
50.0	48.13	56.67	64.61	71.95	78.69	84.82	90.35	95.27	99.60
63.0	55.21	63.75	71.69	79.03	85.76	91.90	97.42	102.35	106.68
80.0	45.78	54.31	62.25	69.58	76.32	82.45	87.98	92.90	97.23
100.0	41.36	48.37	55.18	61.71	67.89	73.66	78.93	83.65	87.73
125.0	57.25	64.26	71.06	77.59	83.77	89.54	94.81	99.53	103.61
160.0	66.09	73.10	79.90	86.43	92.62	98.38	103.66	108.37	112.46
200.0	57.05	64.06	70.87	77.40	83.58	89.34	94.62	99.34	103.42
250.0	17.04	24.06	30.86	37.39	43.57	49.35	54.70	59.69	64.46
315.0	.00	.00	.00	10.38	26.84	41.45	54.21	65.11	74.17
400.0	.00	.00	8.95	27.26	43.72	58.33	71.09	81.99	91.05
500.0	.00	2.01	22.17	40.48	56.95	71.55	84.31	95.22	104.27
630.0	.00	8.48	28.64	46.95	63.41	78.02	90.78	101.69	110.74
800.0	.00	.00	18.81	37.13	53.59	68.20	80.95	91.86	100.91
1000.0	.00	.00	.00	.00	15.50	30.11	42.86	53.77	62.82
1250.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
1600.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
2000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
2500.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
3150.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
4000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
5000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
6300.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
8000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
10000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00

	Acoustic Angles from Inlet, deg.							
	100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0
Freq.								
50.0	103.32	106.44	108.95	110.87	112.18	112.89	112.99	112.49

63.0	110.40	113.52	116.03	117.94	119.26	119.96	120.07	119.57
80.0	100.95	104.07	106.58	108.49	109.80	110.51	110.62	110.12
100.0	91.11	93.72	95.49	96.34	96.21	95.02	92.72	89.23
125.0	107.00	109.60	111.37	112.22	112.08	110.89	108.56	105.04
160.0	115.84	118.45	120.21	121.06	120.92	119.73	117.40	113.88
200.0	106.80	109.41	111.18	112.02	111.89	110.69	108.37	104.85
250.0	68.99	72.89	75.70	77.08	76.89	75.12	71.85	67.24
315.0	81.37	86.72	90.21	91.86	91.65	89.59	85.68	79.92
400.0	98.25	103.59	107.09	108.74	108.53	106.47	102.56	96.79
500.0	111.47	116.82	120.31	121.96	121.75	119.69	115.78	110.02
630.0	117.94	123.29	126.78	128.43	128.22	126.16	122.25	116.49
800.0	108.11	113.46	116.96	118.60	118.40	116.34	112.42	106.66
1000.0	70.02	75.37	78.87	80.51	80.30	78.24	74.33	68.57
1250.0	.00	.00	.00	.00	.00	.00	.00	.00
1600.0	.00	.00	.00	.00	.00	.00	.00	.00
2000.0	.00	.00	.00	.00	.00	.00	.00	.00
2500.0	.00	.00	.00	.00	.00	.00	.00	.00
3150.0	.00	.00	.00	.00	.00	.00	.00	.00
4000.0	.00	.00	.00	.00	.00	.00	.00	.00
5000.0	.00	.00	.00	.00	.00	.00	.00	.00
6300.0	.00	.00	.00	.00	.00	.00	.00	.00
8000.0	.00	.00	.00	.00	.00	.00	.00	.00
10000.0	.00	.00	.00	.00	.00	.00	.00	.00

PROGRAM: COMBUSTOR Version 1.2 17-May-99 10:01:16

GEAE/ADT Combustor Noise Prediction
 - Modified Diversitech Correlation

INFO = Sample DAC Combustor
 IALPHA = SAE77
 TAMB = 77 Deg. F
 PAMB = 29.92 In. Hg
 RELHUM = 70%
 DIST = 1 Ft.
 IARCSL = ARC
 SACORDAC = DAC
 DACTYPE = 20-10
 CLENGTH = .492
 ANNHTH = .25
 AERAEFF = 2.43
 NUMFUEL = 30
 MXFUELNZ = 40
 DIAHYD = 13.4
 DIAEFF = 39.9
 W3 = 113.9
 T3 = 1189.9
 P3 = 173.1
 T4 = 2260.0
 P4 = 162.9
 T8 = 1205.6
 P8 = 15.1
 FFC = .6655269
 FFT = 7.87937
 SPLFFC = -3.53669
 SPLTL = 12.02675
 XK1 = 3.99335
 CP160 = 14.27774
 CP500 = 34.17858
 OASPK160 = 81.19081

OASPK500 = 78.0304

Double Annular Combustor Spectra

Freq.	Acoustic Angles from Inlet, deg.								
	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0
50.0	.00	.00	.00	.00	.72	11.20	20.30	28.02	34.35
63.0	.00	.00	.00	3.41	15.27	25.76	34.85	42.57	48.90
80.0	.00	.00	6.34	19.59	31.45	41.94	51.03	58.75	65.08
100.0	.00	6.60	21.24	34.49	46.35	56.84	65.93	73.65	79.98
125.0	2.93	18.94	33.58	46.83	58.70	69.18	78.28	85.99	92.32
160.0	8.47	24.49	39.13	52.38	64.24	74.73	83.82	91.54	97.87
200.0	.00	14.07	28.71	41.96	53.83	64.31	73.41	81.12	87.45
250.0	.00	.00	.00	13.45	27.70	40.43	51.59	61.14	69.08
315.0	.00	.00	11.88	27.99	42.46	55.30	66.52	76.11	84.07
400.0	.00	6.89	24.62	40.72	55.19	68.04	79.25	88.84	96.80
500.0	.00	11.82	29.55	45.65	60.13	72.97	84.19	93.77	101.73
630.0	.00	1.99	19.72	35.82	50.30	63.14	74.36	83.94	91.90
800.0	.00	.00	.00	.00	9.74	22.58	33.80	43.38	51.34
1000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
1250.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
1600.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
2000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
2500.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
3150.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
4000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
5000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
6300.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
8000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00
10000.0	.00	.00	.00	.00	.00	.00	.00	.00	.00

Freq.	Acoustic Angles from Inlet, deg.							
	100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0
50.0	39.29	42.86	45.04	45.83	45.24	43.27	39.91	35.17
63.0	53.85	57.41	59.59	60.38	59.79	57.82	54.46	49.72
80.0	70.02	73.59	75.77	76.56	75.97	74.00	70.64	65.90
100.0	84.93	88.49	90.67	91.46	90.87	88.90	85.54	80.80
125.0	97.27	100.83	103.01	103.80	103.21	101.24	97.88	93.14
160.0	102.82	106.38	108.56	109.35	108.76	106.79	103.43	98.69
200.0	92.40	95.96	98.14	98.94	98.35	96.38	93.02	88.28
250.0	75.40	80.09	83.16	84.60	84.42	82.61	79.17	74.11
315.0	90.40	95.10	98.17	99.62	99.44	97.63	94.19	89.12
400.0	103.13	107.84	110.91	112.36	112.17	110.36	106.93	101.86
500.0	108.07	112.77	115.84	117.29	117.11	115.30	111.86	106.79
630.0	98.24	102.94	106.01	107.46	107.28	105.47	102.03	96.96
800.0	57.68	62.38	65.45	66.90	66.72	64.91	61.47	56.40
1000.0	.00	.00	.00	.00	.00	.00	.00	.00
1250.0	.00	.00	.00	.00	.00	.00	.00	.00
1600.0	.00	.00	.00	.00	.00	.00	.00	.00
2000.0	.00	.00	.00	.00	.00	.00	.00	.00
2500.0	.00	.00	.00	.00	.00	.00	.00	.00
3150.0	.00	.00	.00	.00	.00	.00	.00	.00
4000.0	.00	.00	.00	.00	.00	.00	.00	.00
5000.0	.00	.00	.00	.00	.00	.00	.00	.00
6300.0	.00	.00	.00	.00	.00	.00	.00	.00
8000.0	.00	.00	.00	.00	.00	.00	.00	.00
10000.0	.00	.00	.00	.00	.00	.00	.00	.00
